パソコンプログラマのためのクリエイティブマガジン

1991年1月1日発行(毎月1回発行) 第1巻 第1号 通巻1号

月刊プログラマーズ・ページ

Programmer's Page

MONTHLY MAGAZINE FOR PROGRAMMERS

1991 1 創刊号 JANUARY

定価980円



特集 プログラマの仕事場

パソコンプログラマの開発環境を徹底レポート

創刊記念プレゼント HANDY98他

SE

Shoeisha Co., Ltd.

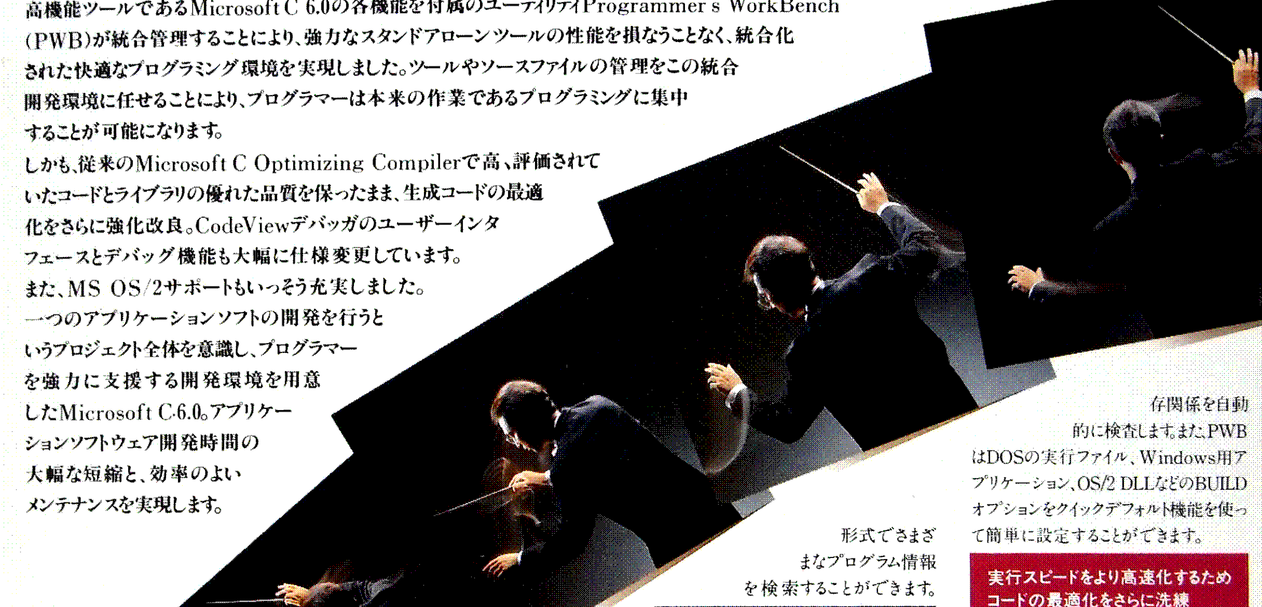
これが、Cの正統。

アプリケーションソフトウェアの開発に携わるプロフェッショナルユーザーの方々のために、強力で柔軟な統合開発環境を提供するMicrosoft C Professional Development System Version 6.0(Microsoft C 6.0)。独立した高機能ツールであるMicrosoft C 6.0の各機能を付属のユーティリティProgrammer's WorkBench(PWB)が統合管理することにより、強力なスタンドアローンツールの性能を損なうことなく、統合化された快適なプログラミング環境を実現しました。ツールやソースファイルの管理をこの統合開発環境に任せることにより、プログラマーは本来の作業であるプログラミングに集中することが可能になります。

しかも、従来のMicrosoft C Optimizing Compilerで高い評価されていたコードとライブラリの優れた品質を保ったまま、生成コードの最適化をさらに強化改良。CodeViewデバッガのユーザーインタフェースとデバッグ機能も大幅に仕様変更しています。

また、MS OS/2サポートもいっそう充実しました。

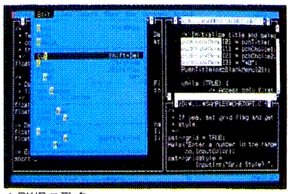
一つのアプリケーションソフトの開発を行うというプロジェクト全体を意識し、プログラマーを強力に支援する開発環境を用意したMicrosoft C 6.0。アプリケーションソフトウェア開発時間の大幅な短縮と、効率のよいメンテナンスを実現します。



PWBを中心に生産性を強力に向上

●強力なエディタ機能

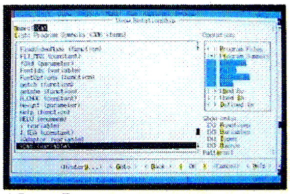
QuickC 2.0と同様のユーザーインタフェースを持つProgrammer's WorkBench(以下PWB)のエディタ機能。しかも、キーの割付変更や編集機能の拡張などの豊富なカスタマイズ機能、独自のマクロ言語を使った編集機能の定義、レコードマクロ、同時に8つのウィンドウをオープンできるマルチウィンドウ機能を装備しています。



▲PWBエディタ

●Source Browser機能

PWBは作成中のアプリケーションソフトウェアの構造を解析するSource Browser機能を備えています。これにより、対話形式でさまざまなプログラム情報を検索することができます。



▲Source Browser

●Microsoft C Advisor

PWBはQuickC 2.0同様に強力なオンラインリファレンスシステムを装備しています。



▲Microsoft C Advisor

●プログラムメンテナンス機能

PWBは、プログラムリストを使って簡単にMAKEファイルを自動作成。プログラムリストは、MAKEファイルを作成する際に、インクルードファイルを含むファイル依存関係を自動的に検査します。また、PWBはDOSの実行ファイル、Windows用アプリケーション、OS/2 DLLなどのBUILDオプションをクイックデフォルト機能を使って簡単に設定することができます。

実行スピードをより高速化するためコードの最適化をさらに洗練

●グローバルな最適化の実現

Microsoft C 5.1で行われていた"ループレベル"での最適化から、関数全体に対してグローバルな最適化を実現するように進化。また、変数を自動的にレジスタ変数に割り当てるようにしています。

●ローカルコードの最適化の実現

数式を手作業で最適化するときの"経験則"を導入。shift式やswitchステートメントを有効に最適化できます。

●最適化レベルの管理の強化

コンパイル時の指定以外にプラグマによっても最適化の管理ができ、特定の関数を思い通りのレベルで最適化することができます。

コードのパフォーマンス向上と開発時間の短縮を同時に実現する新しい言語仕様

●basedポインタ

nearポインタのコンパクトさとfarポインタの機能を合わせ持つポインタです。nearポインタと同様に16ビットのアドレスしか

●以下のシリーズ対応につきましては、直接各社様宛お問い合わせください。

FM Rシリーズ対応(富士通株式会社より発売予定)お問い合わせ先:03-730-3140

Panacom Mシリーズ対応(松下電器産業株式会社より発売予定)お問い合わせ先:0120-158756

ありがとうキャンペーン 実施中

期間:'90年12月1日〜'91年1月31日

この期間にマイクロソフト製品をご購入になり、登録された方の中から抽選で3,000名様に、マイクロソフトグッズをプレゼント。詳しくは、キャンペーンポスターのあるお店で。

LIMは、Lotus/Intel/Microsoftが提唱するEMSの方式です。またEMSはExpanded Memory Specificationの略語です。
ATOKは㈱ジャストシステムの、Lotusは米国ロータスデベロップメント社の、Intelは米国インテル社の登録商標です。
Microsoft, MS, MS-DOS, CodeViewは、米国マイクロソフト社の登録商標です。MS-KANJI APIは、マイクロソフト株式会社が推奨する日本語入力システムの仕様です。



必要とせず、farポインタのように64Kを超えるメモリセグメントの参照を可能にしました。

●プロシージャコールの最適化

新しいプロシージャコール_fastcallは、関数間の引数の受渡しの際に直接レジスタを利用することで実行速度を大幅に向上させます。

●インラインアセンブリ機能

Microsoft C 6.0のコンパイラは、インラインアセンブラを統合しています。もちろん、アセンブルの際には、Microsoft Macro Assemblerは必要ありません。

●TINYメモリモデルの追加

マイクロソフト社の高級言語ではじめてTINYメモリモデルをサポート。このメモリモデルを利用することにより、MS-DOS上で動作するCOMファイルをC言語で作成することができます。(ただし、グラフィックスライブラリ、BIOSライブラリは使用できません)

●ライブラリの強化

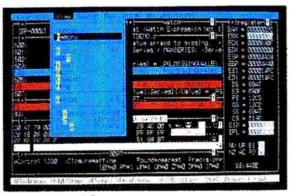
BIOSライブラリと強化されたグラフィックスライブラリを装備。もちろん、FONTライブラリ、チャートライブラリも追加されました。

最新のCodeView 3.0で デバッグ機能を強化し ユーザーインタフェースを改善

●容量の増大

CodeView自身のコードをEMSに追い出すことにより、より大きなターゲットプログラムのデバッグが可能になりました。

●データアクセスの拡大



▲CodeView 3.0

ローカルウィンドウをサポート。また、ローカルウィンドウやウォッチウィンドウを利用し、配列やネスト構造体の内容を展開表示したり、ポインタが指しているアドレスの内容を表示したりすることが可能です。さらにデバッグ中の8087/80287/80387コプロセッサのステータスを表示するためのコプロセッサウィンドウや、指定した変数や式の現在の値を即座に表示するクイックウォッチ機能が追加されました。

●ユーザーインタフェースの改良

ユーザーインタフェースを完全に再設計し、QuickシリーズやPWBと同様のユーザーインタフェースを実現。各ウィンドウは独立したまま共通の操作環境を持ち、各ウィンドウ単位での縦横のスクロールや内容の印刷も可能です。

●ダイナミックリプレイ機能

デバッグ途中のプログラムに変更を加えた場合、変更した時点までに入力したすべての情報をプログラムの先頭から自動的に再生するダイナミックリプレイ機能を装備しています。

●PWBとの統合

CodeView 3.0は、従来のCodeViewと同じく独立したツール。しかし、PWBとの組み合せを考えた設計を行い、PWB側であらかじめ設定しておくことが可能です。また、CodeView3.0のデバッグセッションの情報も自動的に保存されますから、いったんPWB側に戻った後でもウィンドウの設定、ウォッチ変数、ブレークポイントなどの情報は自動的に再設定されます。

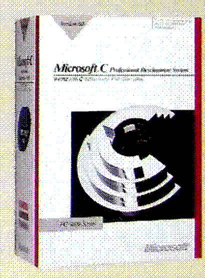
■必要システム

[NEC PC-9800シリーズ PC-98シリーズ]

- ●ハードウェア:80286、80386をCPUに持つNEC PC-9800シリーズ、またはPC-98シリーズ(ただし、ハイレゾリューションモードを除く)
- ●OS:MS-DOS 3.1以上、もしくはOS/2 1.1A以上
- ●必要メモリ:640Kバイト(MS-DOS)、空きメモリ1Mバイト以上(OS/2)
- ●ディスク装置:1Mバイト容量の1台以上のフロッピーディスク装置と1台以上のハードディスク装置(ハードディスクの空き容量は最低約6Mバイト以上、最大20Mバイト必要)
- ●推奨システム:1Mバイト以上のLIM4.0対応のEMSメモリ(MS-DOS)、1Mバイト以上のキャッシュディスク(MS-DOS)、マウス
- ●使用可能な日本語入力システム:MS-KANJI API仕様の日本語入力システム(ATOK6、ATOK7の場合、付属のドライバで対応可能)

■実行ファイルのターゲットシステム

- ●ハードウェア:NEC PC-9800シリーズ、PC-98シリーズ(ただし、グラフィックスライブラリを使用する場合は、PC-98LTおよびハイレゾリューションモードを除く。またBIOSライブラリを使用する場合は、PC-98LT/XAを除く)
- ●OS:MS-DOS 2.11以上、もしくはOS/2 1.1以上 Microsoft Windows 3.0以上 (実行ファイル作成には日本語MS-WINDOWSツールキットが必要。MS-DOSとMS OS/2のBOUND実行ファイルを作成するときは、日本語MS OS/2ソフトウェア開発支援ツールが必要)
- ●必要メモリ:システムと作成プログラムに依存



Microsoft® C
Professional Development System Version 6.0

NEC PC-9800シリーズ対応 標準価格 98,000円(消費税別) 好評発売中

マイクロソフト株式会社 〒160 東京都新宿区西新宿7-5-25 Kビルディング 電話(03)363-1201

<資料請求番号P02>

Microsoft

使いやすい統合環境と高度で多彩な機能。Quickシリーズは、プログラミン

Microsoft QuickBASIC Ver.4.5は、アプリケーション感覚の自由自在な操作性とパワフルな言語仕様をもった最新のBASIC言語。Quickシリーズならではの使いやすい環境とプロユースにも耐える高度な機能を搭載。他のBASICとは根本的に異なる新世代言語です。
もちろん、初めて言語を学ぶ方もプログラミング言語の基本学習ツールとして活用していただけます。また、構造化とモジュール化の強力なサポートにより、小回りのきくユーティリティの作成から高度なアプリケーションの開発まで、プロフェッショナルの方にも、即戦力としてご利用いただけます。

アプリケーション感覚で使える統合環境

Quick BASICの統合環境はエディタ、インタープリタ、コンパイラ、デバッガをすべて標準装備。エディタのドロップダウンメニューから、全ユーティリティを実行できます。基本メニューは、利用頻度の高いコマンドだけを集めたEasyメニューと、全機能を網羅した高機能なFullメニューの2種類を提供。また、プログラム入力以外のすべての操作はマウスでOK。エディタを高速化し、カーソル惰性も抑止しました。


●ドロップダウンメニュー

入力ごとに構文チェック

プログラムを一行入力するごとに構文チェック。文法的な誤りなどに対しては、即座にエラーメッセージを表示します。イージーミスがすぐに発見でき、さらにスムーズなプログラム開発を実現します。

モジュール化の概念を導入

サブルーチン、ユーザー関数の考え方を大幅に拡張。プロシージャの概念を導入し、プログラムをモジュール単位で管理。1度作ったサブモジュールは、他のプログラムに再利用できます。また、MS-CやMASM、MS-FORTRAN、QuickCで作成したプロシージャを呼び出し、関数やステートメントのように利用することもできます。

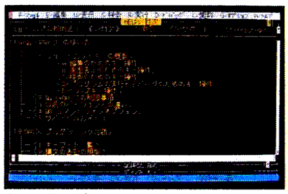
＊Microsoft C 6.0には対応していません。

構造化プログラミングをサポート

構造化のための制御文をフルサポートし、構造化プログラミングを強力にバックアップ。BASICを行番号から解放し、スパゲティ・プログラムを追放します。

初心者にやさしい学習環境

QBチュートリアル： QuickBASICの画面上の名称や構造化言語・モジュール化の概念などを学べます。マニュアルにはプログラム例をまじえた解説やプログラム作成の工程が分かりやすく説明されています。
QuickBASICプログラミングガイド： サブプログラムの作り方やデバッグの方法などプログラミングの基本的な流れが、サンプルプログラムを完成させる形で学べます。
QBアドバイザ： オンラインヘルプ機能を飛躍的に進化させた強力なQBアドバイザが、統合環境の操作方法、各関数やステートメントの詳細な説明、プログラム例、エラーメッセージの解説、操作方法の説明などをスピーディーに表示します。いちいちマニュアルを読む必要はありません。


●QBアドバイザ

強化・拡張されたステートメントと関数

PLAY*、SOUND*、SLEEP、UEVENTステートメントを追加。また、PRINT、LOCATE、CIRCLEステートメントの処理を高速化。Ver.4.2のステートメントや関数も強力な言語仕様へ拡張。実行ファイルの処理速度は非常に高速です。

＊PC-9801E/F/M/Uでは利用できません。また、PLAYステートメントはFM音源ボードには対応していません。

多彩なデバッグ機能

ブレークポイントを設定して実行、変数の内容を見ながらトレース、一定条件を満たすまで実行など、多彩なデバッグ機能を備えています。

高機能インタープリタと高速コンパイラ

編集画面と実行画面を独立して管理する統合環境を持ったインタープリタと、QuickBASICがなくてもMS-DOS上で動作する高速なEXEファイルを作成するコンパイラを標準で装備。いずれも数値演算コプロセッサに対応しています。

パス設定コマンドの追加

メタコマンドで読み込まれるインクルードファイルやライブラリ、実行ファイルなどのパス設定が行えます。

実用的なライブラリ

マウス対応のプログラムを作るためのプロシージャ群や、漢字入力システムのオン/オフ制御、グラフィック画面のスクロールやグラフィック画面への文字描画、画面ハードコピールーチンなど、実用的なライブラリを用意。ソースプログラムも付属しています。

■基本システム
〔NEC PC-9800シリーズ〕
●本体NEC PC-9800シリーズ(ただし、PC-9801、PC-98XA/LTおよびXL/XL²/RLのハイレゾリューションモードを除く)
●日本語MS-DOS Ver.2.0以上
●必要メモリ：640Kバイト
●利用可能な日本語システム：MS-KANJI API仕様のもの。もしくはATOK6。
●ハードディスク、マウスのご利用をお勧めします。



Microsoft® QuickBASIC

Version 4.5

NEC PC-9800シリーズ対応　標準価格20,000円(消費税別)　好評発売中

Quickが



グ言語の学習から本格的アプリケーション開発まで、強力にサポートします。

パソコン用プログラム開発の主力言語として、いま最も人気の高いC言語。しかし、多数のユーティリティが必要、いちいちマニュアルを読まなくては使いこなせない、動作確認の度に実行ファイルが必要、論理的な誤りをおかしがちな文法、膨大な関数を覚えなくてはならないなどの点で、初心者には難しいといわれているのも事実です。
でも、QuickC2.0なら大丈夫。これらの問題点を解決し、初心者の方にもやさしいプログラミング環境を提供します。しかも、従来バージョンに比べて機能は飛躍的にアップ。C言語の学習からプロユースまで、最良のCコンパイラとしてご活用いただけます。

プログラム開発を容易にする統合環境

エディタ、コンパイラ、リンカ、デバッガを統合環境上で利用可能です。起動するとエディタの状態になり、すぐにプログラムが書けます。書き終わったら、後はファンクションキーを押すだけ。コンパイル&リンク&実行は自動的に行われます。もちろん実行ファイルも自動作成します。

初心者にやさしい学習環境

QCアドバイザ：オンラインヘルプ機能を強化改良したQCアドバイザが、場面に応じて適切にアドバイス。操作方法、ランタイム関数やキーワードに関する解説が、編集しながら呼び出せます。しかも、コンパイル中に発生したエラーの説明もすぐに参照できます。また、ランタイムライブラリ関数の説明には、すべてサンプルプログラムが付いています。さらに、付属のHELPMAKEユーティリティを利用して、標準のヘルプファイルを作り直したり、オリジナルのヘルプファイルを作成することも可能です。



●QCアドバイザ

CBT：対話形式のComputer Based Training（CBT）で、統合環境の使い方、プログラムの作成、デバッグの方法などをコンピュータ上でスピーディーに学習できます。

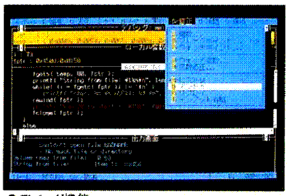


●CBT

C for Yourself：初心者の方のための学習マニュアル「C for Yourself」が付属。難しいC言語の文法を、豊富なサンプルプログラムをもとに分かりやすく解説してあります。しかも、サンプルプログラムは、QCアドバイザから簡単に呼び出せます。

強化されたデバッグ機能

Quick C 2.0は、強力なデバッグ機能を内蔵します。ローカルウォッチ変数や実行画面をウィンドウで表示できます。さらに、レジスタウィンドウもサポートします。また、Quick C 2.0のデバッグ機能は、同一の環境上で実現されるので、作成したプログラムをコンパイル&リンク後、そのソースプログラムを表示したままトレースやブレークポイントの設定をすることが可能です。



●デバッグ機能

やさしいインタフェース

メニュー選択や編集操作などほとんどの操作がマウスで行えます。また、編集キーでの編集方法は、Microsoft WindowsやMicrosoft Worksと共通なので、とても簡単。覚えた操作はこれからのマイクロソフト製品でもそのまま利用できます。

豊富なユーザーカスタマイズ機能

編集キーの割り当てを変更するMKKEYユーティリティ、画面の表示色を自由に変更できるQCCONFIGユーティリティを追加。さらに、ユーザーによるメニューの追加を実現するカスタマイズメニュー機能も装備されます。

充実したライブラリ

実数座標をサポートした"グラフィックスライブラリ"、グラフやチャートが簡単に書け、多種類のフォントが揃った"チャート/フォントライブラリ"、BIOSコールを直接サポートする"BIOSライブラリ"、MS-KANJI API仕様の日本語フロントエンドプロセッサをアプリケーションから制御する"カナ漢サポートライブラリ"を追加しました。

NMAKEユーティリティ

UNIX上のMAKEユーティリティに近い機能を持つ最新のNMAKEユーティリティを標準装備しています。統合環境上でNMAKEが利用可能なMAKEファイルも自動作成することができます。

インクリメンタルコンパイル&リンクを実現

コンパイル&リンクがさらに高速に。しかも統合環境上でコンパイル可能なソースファイルのサイズも飛躍的に拡大しました。

基本です。



■基本システム
〔NEC PC-9800シリーズ〕
●本体NEC PC-9800シリーズ（ただし、XA・LTおよびXL・XL^2・RL・H98のハイレゾリューションモードを除く）
●日本語MS-DOS Ver2.0以上
●必要メモリ：640Kバイト
●利用可能な日本語システム：MS-KANJI API仕様のもの。もしくはATOK6。
●ハードディスク、マウスのご利用をお勧めします。



Microsoft® QuickC®
Compiler Version 2.0

NEC PC-9800シリーズ対応　標準価格20,000円（消費税別）　好評発売中

マイクロソフト株式会社
〒160 東京都新宿区西新宿7-5-25 Kビルディング 電話(03)363-1201

〈資料請求番号001〉

BORLAND

1990年。TURBO Cは、進化の第2ステップへと踏み出した。

TURBO CからC++へ――。
話題のオブジェクト指向プログラミング(OOP)言語
C++ & Cコンパイラ、ボーランドからいよいよ新登場。



C++は、90年代の主流となるオブジェクト指向プログラミングを、自然な形でC言語に取り入れた言語です。TURBO C++は、"TURBO"の速度と使いやすさをC++(AT&T C++ Ver.2.0に準拠)で実現しました。もちろん、ANSI準拠のCコンパイラとしての機能も備えており、既存の資産をムダにすることはありません。*

*プログラムによっては、再コンパイルしなければならない場合があります。

■Programmer's Platform

統合環境が大きく改善されました。オーバーラップマルチウィンドウをサポートした環境は、アプリケーション開発に必要な編集・コンパイル・リンク・デバッグ機能を備えており、マウス操作も可能です。エディタは、複数ファイルを同時にオープンすることができ、編集するファイルの大きさはディスク容量にのみ制限されます。キャッシュ機能による高速処理で大規模なプログラム開発にも対応でき、分割コンパイルのサポートも万全です。トランスファ機能では、TURBO ASSEMBLERやTURBO DEBUGGERなどの外部プログラムを簡単に呼び出すことができます。TURBO ASSEMBLERのエラーメッセージは、統合環境にも取り込めます。

■VROOMM

VROOMM(Virtual Runtime Object-Oriented Memory Manager)は、ボーランドの進化したオーバーレイマネージャです。オーバーレイ化されたコードは、必要なとき自動的に、しかも効率よくオーバーレイバッファに呼び出されます。プログラマは、オーバーレイを意識する必要はほとんどありません。

■拡張されたさまざまな機能

プログラムの中でのコンパイルオプションの記述、大規模な静的データの確保、多くのファイルを一度にコンパイルするための応答ファイルのサポートなど、細かい部分でも数多くの点が改良されました。

■充実したユーティリティ群

関数毎にサンプルプログラムを備えたヘルプユーティリティ、自己スワップや応答ファイルの自動生成をサポートしたMAKEユーティリティなど、プログラム開発に必要なユーティリティも揃っています。

■ハイレゾルーションモードに対応

統合開発環境のハイレゾモードでの動作、31行モード、BGIやテキストライブラリのハイレゾモードをサポート。また、TURBO DEBUGGER & TOOLSもバージョン2.01**としてハイレゾモードに対応しました。

**TURBO DEBUGGER & TOOLS 2.0の登録ユーザ様には、新Ver.2.01をお送りします。

■TURBO C++ 1.0の動作環境

機種：PC-9801シリーズ、PC-286シリーズ(ハイレゾモード対応、LTは除く)／メモリ：640KB以上／MS-DOS：バージョン2.11以上／マウス：バスマウス

新発売記念キャンペーン実施!

●キャンペーン期間中(11月末～'91年3月末)に限り、特別価格での販売を実施いたします。

TURBO C++ 1.0…………39,800円→29,800円
TURBO C++ Professional…59,800円→49,800円

TURBO C 2.0…………19,500円(11月1日～)

●登録ユーザー様にはアップデートのご案内をお送りいたします。まだ登録の済んでいない方は、お早めに登録カードをご返送ください。

※製品の価格には、消費税は含まれていません。

新発売

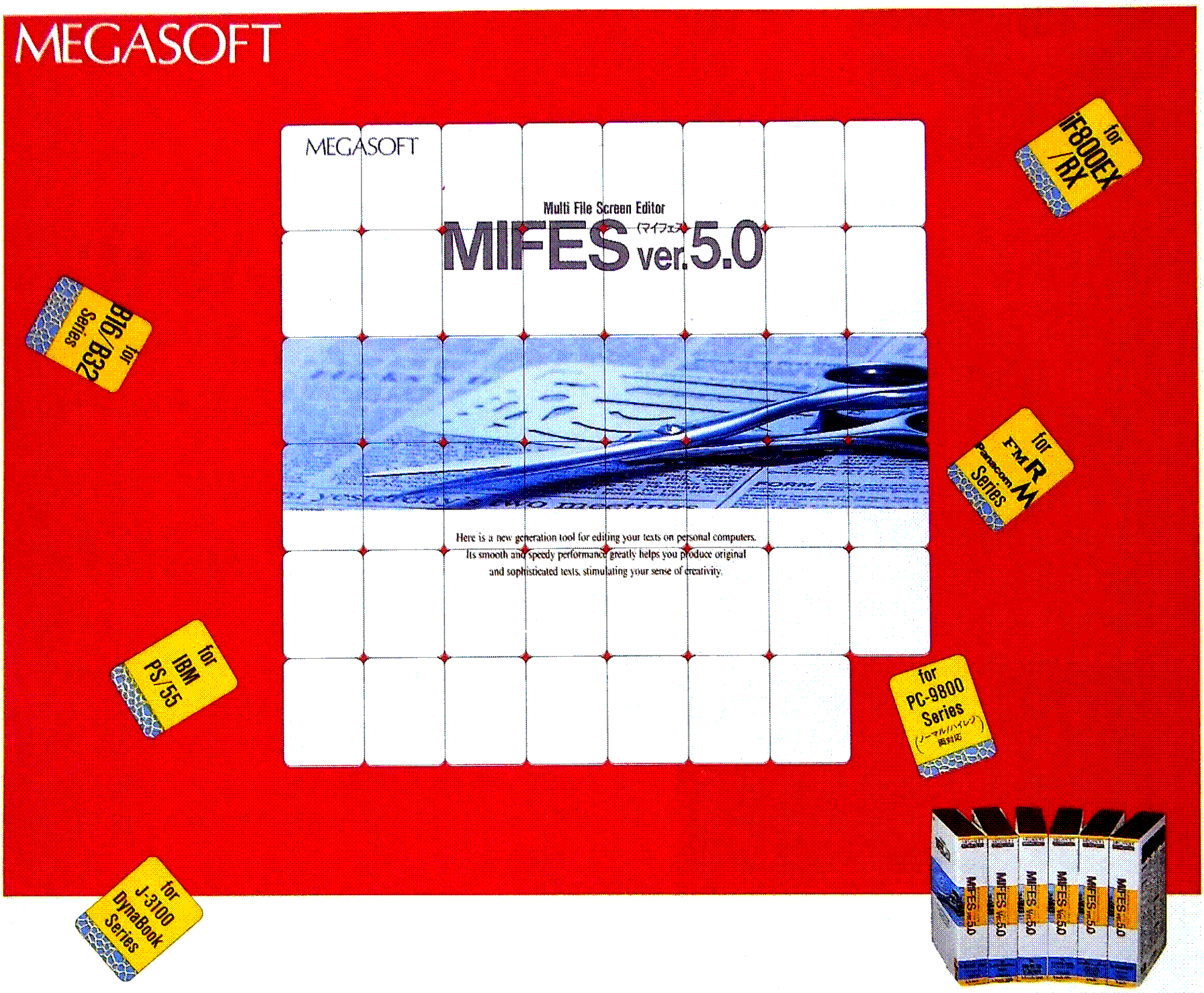


TURBO C++® PROFESSIONAL

●製品に関するお問い合せは、総販売元㈱マイクロソフトウェアアソシエイツ TEL.03-486-1411までお願いします。

株式会社ボーランドジャパン
〒107 東京都港区南青山7-8-1 小田急南青山ビル TEL.03-486-1400㈹

〈資料請求番号002〉



対応機種の豊富さがスタンダードの証しです。

■EMSをサポート。■ファイルサイズ約46KBの小型軽量エディタMIFES-miniを添付。■ハイレゾ/ノーマル両モードに対応(PS/55, J-3100版を除く)。■日本語DOS/英語DOS両モードに対応(PS/55, J-3100版)。■表作成時に便利な箱型編集機能・罫線機能。■ワープロ感覚で自由にカーソル移動できるフリーカーソルモード。■同時に最大15個までのファイルをオープン可能。■日本語入力FEPに自動対応し簡易ワープロとしても使用可能。■Ver.4で好評の高機能マクロ言語MILをさらに強化して搭載。■編集を終了せずに印刷のできる印刷ユーテイリティーMIPPが付属。■ファイル選択時にサイズや日付も確認できる拡張表示形式を追加。■ヘルプ画面から機能を選択しダイレクト実行が可能。■ファイル名入力時にカーソル移動による行内編集機能をサポート。■日本語対応の正規表現を用いて検索や置換が可能。■画面色やメニュー項目など、カスタマイズ機能を大幅に強化。■OS/2版MIFES/2をはじめ、各機種での操作性を統一。■外部マクロコマンドのサイズを最大4Kバイトまで拡大。■編集可能な1行の長さを最大1500バイトまで拡大。■タグジャンプ機能において、Turbo-Cのエラーフォーマットを追加。■環境変数MIFESにより動作環境を設定可能。■リピートキーへの割り当て状況を常に画面表示。■全角の文字種の変化点をデリミターとして認識。■MIFIND.EXEの検索速度を約5倍にスピードアップ。■作成したマクロ機能を任意のキーに割り当て可能。■実用的な外部マクロコマンドを豊富に標準添付。●視認性と応答性を両立するシフトスクロール機能。●連続したキー入力にも充分応答できる高速応答性。 EOFコードを文字として扱うモードを用意。●可変タブ、行マーク、オートインデントなど多彩な機能を用意。●高機能な行/文字列/箱型のCUT&PASTE機能をサポート。●強力なヘルプ機能や2行ガイド機能など、初心者向け配慮も万全。■前回の終了時の状態から編集を再開するプロファイル機能を強化。●日本で最大のシェアを持つパソコン用標準エディタ。■50種以上の起動時オプションで多彩なニーズに対応。

※■は新機能、●は従来からの機能

マルチファイルスクリーンエディタ
MIFES 〈マイフェス〉 Ver.5.0

■FMR/Panacom M用(ハイレゾ/ノーマル両対応版)……¥48,000
■B16/B32シリーズ用(ハイレゾ/ノーマル両対応版)……¥48,000
■if800EX/RX/BS21用(ハイレゾ/ノーマル両対応版)……¥48,000
■PC-9800シリーズ用(ハイレゾ/ノーマル対応版)……¥38,000
■J-3100シリーズ用(日本語/英語モード両対応版)……¥38,000
■PS/55用(日本語/英語モード両対応版)……¥38,000

使いやすさと信頼性――
メガソフト株式会社 〒564 吹田市江の木町1-38西谷東急ビル
Tel.06-386-2043 Fax.06-386-2123

〈資料請求番号003〉

プログラミングブックス

マイクロソフトQuickシリーズのスペシャルリファレンス



Quick BASIC Ver. 4,5
ステートメント&関数リファレンス

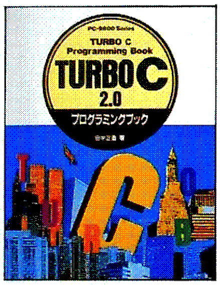
Quick C コンパイラVer. 2,0
ランタイムライブラリリファレンス



●マイクロソフト株式会社 監修 ●定価各2,400円

マイクロソフトの超人気言語ソフトQuick シリーズ。Quick BASIC Ver. 4, 5 Quik C Ver. 2,0スダンダードリファレンス。マイクロソフト社監修でオンラインヘルプを完全書籍化サンプルプログラム満載のリファレンス。

初級〜中級プログラマーのための実践ガイダンス



Turbo C Ver. 2,0
プログラミングブック

●田中正造 著 ●定価2,000円

BGI／ROM BIOS/EMS の実践テクニックガイダンス、Turbo C 2,0の実徴的な機能を豊富な図表と実践有なサンプルプログラムで解説。



Quick BASIC Ver. 4,5
プログラミングブック

●塚本吉彦 著 ●定価2,400円

Quick BASIC で気軽にプログラミング。BASICビギナーだけではなく昔からのBASIC ファンにも贈るQuick BASIC の入門書です。

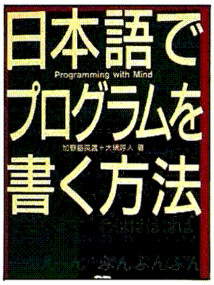
日本語プログラミングへの招待



日本語プログラミング言語
MIND

●片桐明 著 ●定価2,884円

日本語でプログラムが書ける画期的なソフト「Mind」。開発者自身の執筆による決定版解説書。



日本語でプログラムを
書く方法

●加野島英渡・大網呼人 著
●定価1,200円

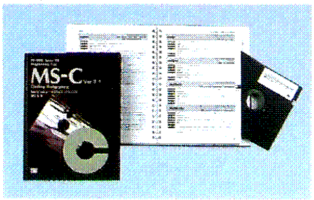
本書には、記号、数式は一切出てきません。でもプログラミングの本なのです。日本語で書いたプログラムが本当にパソコンで動きます。

ディスク付MS-Cリファレンス

MS−C Ver. 5,1
オンラインリファレンス

●谷口崇 著 ●定価2,200円

フロッピーディスク付のプログラミングツール。



エッセイ

✣読書するプログラマ ●祐安重夫 著 ●定価1,648円

「インターフェイス」誌連載。博学でならす著者による読書日記。

✣デジタル版徒然草 ●園田康篤 著 ●定価1,545円

「ＰＣワールド」誌連載。一流のプログラマが世相を斬る。

※定価はすべて消費税込みとなっております。

SE 株式会社 翔泳社
東京都千代田区平河町2-4-14 平河町KSビル 〒102
PHONE 03-263-0447 FAX 03-263-4526

〈資料請求番号004〉

Brief 日本語

■Full Screen Editor

IBM PCでトップセールスに輝く
プロフェッショナル エディター「BRIEF」

ブリーフの高機能
その①

■「メモ」「書式」「スケジュール」etc……

編集中も表示可能、マクロの応用機能。



メモ画面



対応機種一覧

マクロは「工夫」を実現します!

たとえば、マクロでオーバーラップ・ウインドウを表示することができます。表示だけでなくウインドウの中に書き込むこともできます。もちろんアンドウも使用できます。これを使えばいろいろなことができそうです。
編集中に電話番号やちょっとしたメモが必要になったりします。そのたびに住所録やメモ帳を取り出すのは面倒なものです。そんなときはオーバーラップ・ウインドに表示させましょう。メモを見たり、書き込んだり、机の上を探し回る無駄な時間は最小限にしたいものです。

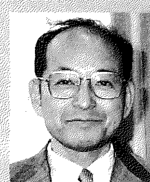
使わない方にも便利なマクロです!

「自分で作らないからマクロ機能はいらない!」と決めつけないでください。公開されたマクロを手にいれれば新しい機能を使うことができるのです。作らなくても機能を増やせることになります。このため弊社では、公開マクロのユーザー相互利用を推進しております。
Wordstar用のキー・アサインや行のソート処理など魅力的なマクロを公開しています。
Briefマクロプログラミング・コンテスト(8月末締切)の応募マクロも公開する予定です。公開マクロについてはBriefユーザー・サポート係にお問い合わせください。

USER'S VOICE

Briefとは長い付き合いになると思う。

私どもの実験は非常に厳しい条件で行うためデータの記録には迅速かつ正確さが要求されます。記録に漏れがあればデータの分析ができなくなり、全ての努力が無駄になります。このためマクロを利用し、記録項目をガイドに従って入力させることにしました。また、入力ミスにはアンドウで対処することで、迅速かつ正確な入力が実現できました。
日本語Briefとこれから先は長い付き合いとなると思います。



東京理科大学 工学部経営工学科
富澤儀一助教授

●革命的マルチマシン・エディター

東芝・NEC・AX協議会・IBM・富士通の5機種のパソコンで動作します。

●300回ものアンドゥ

各ファイルごとに300までの操作を元に戻すことができますので、やり直しの面倒を気にせず、安心して編集できます。

●55ウィンドウ、無限のファイル数

ウィンドウは縦横自由に55まで開くことができます。また同時に編集できるファイルの数や大きさには制限がありません。

●プログラミング

◆自動コンパイル、エラージャンプ◆スワップアウト◆自動インデント、自動テンプレート

●安全編集

Briefは、一定時間キーを操作しない場合、その時点の編集内容をファイルに自動保存しますので、編集中に電源が切れても安心です。

オピニオンリーダー誌のこの評価

- プログラミング向きのエディターを探している人は、もう探す必要はない。Jerry Pournelle(BYTE)
- Briefはプログラマーに最適なだけでなく創造力をかきたてる何物かがある。Computer World
- ほとんど完ぺきにカスタマイズできるだけの拡張性がある。Computer language
- まるで夢のようなアンドゥだ　おもわず間違うのを期待してしまいそうだ。The C Users Journal

■主な機能

◆バッファ書出/切換/削除/リスト表示◆ファイル名の指定/表示/変更◆バックアップ保存◆特殊文字入力◆2行を1行に連結◆ファイル単位の挿入◆増分検索◆各種マーク/始・終点の入換◆ブロックのカット/コピー/ペースト/印刷/保存/左右へのシフト◆対応する括弧へ移動◆C言語のコメント化/非コメント化◆DOSコマンドの実行◆キー操作記憶/実行◆入力ヒストリー◆英文テキスト整形◆タブのスペース挿入◆拡張子毎に異なった設定ができるタブ◆検索置換パターンの画面からの入力◆英字の大小文字変換 その他



定価 48,000円（消費税は含まず）

■全機種対応パッケージ

5"2DD 3.5"2DD一括提供(MS-DOS3.1以降対応)
上記以外のメディアは注文販売となります。
英語用DOSでは使用出来ません。

日本システム株式会社

●お問い合せは 東京都立川市曙町1-18-2一清ビル立川事業所Briefユーザー・サポート係
TEL.0425-23-4469 FAX.0425-27-6127

Programmer's Page
MONTHLY MAGAZINE FOR PROGRAMMERS

パソコンプログラマのための
クリエイティブマガジン
月刊プログラマーズ・ページ

1991 JANUARY No.1

CONTENTS

特集
プログラマの仕事場


Illustration:NAOKI SATO

CONTENTS

表紙イラスト：日暮修一
表紙AD：佐々木幹夫(SE)

広告索引　　資料請求番号

人材募集広告

information・page 1

●イベント

■1月8日(火)〜1月10日(木)

第32回プログラミング・シンポジウム
ソフトウェア全般に関する講演。
会場：箱根ホテル小涌園／参加費：大学院生23,000円（26,000円），
大学助手以上，研究関係者30,000円（35,000円），
会社関係者35,000円（40,000円）予稿集代・宿泊費・食費含
（ ）内は非会員参加費
問い合わせ先：（社）情報処理学会　TEL 03-505-0505

■1月23日(木)〜1月26日(土)

第6回ADEEジャパン'91
IC設計のための最新技術，機器を展示。
第8回エレクトロテスト・ジャパン'91
検査・試験装置の最新技術，機器を展示。
第20回インターネプコン・ジャパン'91
プリント回路基盤作成のための最新技術・関連機器を展示。
第11回国際レーザー／アプリケーション'91
理化学・産業分野のレーザー関連製品，技術を展示。
会場：幕張日本コンベンションセンター／入場料金：3,000円
問い合わせ先：カーナーズエクスポジションジャパン（株）　TEL 03-349-8501

■1月30日(水)〜2月1日(金)

pege '91
「インテグレーテッド・ページネーション」をテーマに，
エレクトロニック・プリプレスシステム，コンピュータ・パブリッシングシステムを展示
会場：池袋サンシャインシティコンベンションセンターTOKYO／入場料金：500円
問い合わせ先：（社）日本印刷技術協会　TEL 03-384-3111

●セミナー

Microsoft University

■1月21日(月)〜1月22日(火)
■2月5日(火)〜2月6日(水)

Microsoft C入門
実習をまじえたC言語の基本を解説。
会場：JRパソコンステーション上野セミナールーム
受講料：55,000円（2日間）

■2月4日(月)〜2月8日(金)

Microsoft Windows Programming Environment
Windowsアプリケーション開発のための入門コース。
会場：インテルジャパン東京トレーニングセンターMUセミナールーム
受講料：150,000円（5日間）

■1月21日(月)〜1月25日(金)

Microsoft LAN Manager Applications Programming
LANマネージャ用アプリケーション開発のための上級コース。
会場：富士ソフトウェアMUセミナールーム
受講料：180,000円（5日間）

■1月22日(火)〜1月25日(金)

MS OS/2 Programming Environment
OS/2アプリケーション開発のための基本コース。
会場：インテルジャパン東京トレーニングセンター
MUセミナールーム
受講料：120,000円（4日間）

■1月28日(月)〜2月1日(金)

MS OS/2 Applications Programming
OS/2のAPIを使った実習中心のコース。
会場：インテルジャパン東京トレーニングセンターMUセミナールーム
受講料：150,000円（5日間）

MS OS/2 Presentation Manager Programming Environment
OS/2 PMアプリケーション開発のための入門コース。
会場：富士ソフトウェアMUセミナールーム
受講料：150,000円（5日間）

■2月4日(月)〜2月8日(金)

MS OS/2 Presentation Manager Applications Programming
OS/2 PMアプリケーション開発のための上級コース。
会場：富士ソフトウェアMUセミナールーム
受講料：150,000円（5日間）

Microsoft Universityは，OS/2，Windowsアプリケーション開発者のために定期的に開催されているセミナーです。
問い合わせ先：富士ソフトウェア（株）
MU教育センター　TEL 03-452-7722

■1月25日(金)

C言語による画面入出力講座
C言語用画面入力設計支援ツール＆ライブラリ「PP II」を題材に画面設計の重要性などを解説。
会場：東京YMCAホテル／参加費：20,600円
問い合わせ先：（株）ライフボート
C言語による画面入出力講座係
TEL 03-293-4716

NEWS & NEW PRODUCTS

1 データショウ'90開催
ノート・ブック型全盛，さらに小型化と差別化への動き

18回目を迎えるデータショウが，1990年10月22日から25日の４日間，東京国際見本市会場（晴海）で開催された．

米国のCOMDEX SPRING，COMDEX FALLのように，春のビジネスショー，秋のデータショーがパーソナルシステムの主要イベントと位置づけられているが，最近では春のビジネスショーのほうがにぎやかなようだ．出品される製品は，開催前におおかた発表されてしまうため，情報としての新鮮さは乏しいが，各メーカーのマシンが並んでいるのを見ることで，製品の傾向を実感することができる．たとえば，90年のCOMDEX SPRINGでは，80486マシンがいっせいに出品され，たしかに時計の針が少し進んだという印象を与えてくれた．

そうした意味では，今回のデータショウは各メーカーともノート・ブック型パソコンを前面に出してこじんまりまとまってしまった印象が強い．東芝ダイナブックの成功以来，続々と発売されたノート・ブック型パソコンがずらりと並んだ．

その中にあって，ノート・ブックより小さいコンピュータの登場が目をひいた．ひとつは，日本電気のHANDY98ことPC-98HAだ．このLT互換のA5サイズマシンは，ROM DOSを搭載しMS-WORKSバンドルで198,000円だが，もはや１年前の東芝ダイナブックほどのインパクトはない．注目を集めたのは，HANNDY98をもって踊っていたコンパニオンのほうかもしれない．

もうひとつの超小型コンピュータは，京セラのRefaloだ．ニューコンセプトマシンというにふさわしく，B6判大のリフィルタイプに作られた16ビットROM DOSコンピュータに手書き入力などの機能が盛り込まれている．このブースに集まった人は，「どう使う，どう入力するのか」を知ろうと熱心に説明を聞いていた．コントやダンスなど，ショウアップが進むなかでこうした雰囲気のほうがめずらしくなってしまった．486搭載などの次世代マシンはまだちらほらとしか見あたらない．ハイエンドマシンは，マルチメディアやLANというテーマづけをした出し方をしている．

開催前に評判を呼んでいたものに，マッキントッシュの新機種がある．発表時点ではもっぱらその価格が話題になったのだが，実機を見るとデザインが新鮮に感じられる．とくにLCとⅡsiは，CRTとのマッチングがみごとだ．

NeXTも新マシンを登場させた．米国で薄型NeXTstationとカラー版NeXTとが発表されたのは９月のことだったが，その後雑誌などではカラーNeXTの画面を見る機会が少なかっただけに注目を集めていた．

日本での登場時期が取りざたされるWindows 3.0は，富士通と三菱電機がFM RとMAXY386でそれぞれデモンストレーションしていた．PMやWindowsでビデオ動画をウィンドウ表示させて壁面いっぱいに展示．マルチメディアのデモビデオも配布するなど，マルチメディアに力を入れたのは日本アイ・ビー・エムだ．



◀カラー版NeXTの展示で盛況だったキヤノンブース

▼会場となった東京国際見本市会場（晴海）



ワンラインNEWS
富士通ゼネラルは富士通の 32 ビットパソコン「FM TOWNS」用のCD-ROMソフトのテストディスクを編集できる「CD-ROMライター」を開発，販売する．

NEWS & NEW PRODUCTS

2 COMDEX／Fall'90開催（11月12～16日）出展社数，入場者数ともニューレコードを達成！



第12回目を迎えたラスベガスCOMDEX Fall'90は，あらゆる面で過去最大のコンピュータイベントとなった．出展社数は，予想をはるかに上回わる1,860社に達し，入場者も初日から過去を大幅に上回る人出となっている．

コーポレートユーザーの増加に伴い，入場者のほとんどはスーツ族で占められるようになってきたが，人気のブースは身動きがとれないほどの混雑ぶりであり，まさに世界規模のPCビジネスを象徴している．

キックオフを宣言したのはマイクロソフト会長ビル・ゲイツ氏．9時半からはじまった彼のスピーチを聞くために，早朝6時から並ばなければ会場に入れないという超人気ぶりであった．「Information at Your Fingertips!」と題し，今後数年間のうちにコンピュータをもっと普通の人が使えるものにしなければならないことなどを熱心にスピーチした．

今回のCOMDEXの特色は，革新的なキーテクノロジーによるブレイクスルーがあいついで登場し，われわれの目の前に提供されていることにある．そのおもな流れとして次のような点をあげることができる．

- Windows 3.0アプリケーションの爆発的な増加によるGUIの定着
- 486マシンの強力なパワーによるスーパーサーバーの登場
- WindowsとDVIによるマルチメディア環境と動画のサポート
- ノート型コンピュータのゴールドラッシュ的ともいえる急激な台頭
- 上記のブレイクスルーをベースとした全く新しいタイプのアプリケーションの登場

実際に会場では驚くほどの数のブースでWindowsによるアプリケーションのデモを行っていた．ハードウェアメーカー，ソフトウェア業界ともメインのプラットフォームがDOSからWindowsに変わったことを認識している雰囲気である．

90年代の幕開けをGUIが飾ったという意味で，これからのPCビジネス業界はGUIによる全く新しいシーンが展開されていくことを強く印象づけたCOMDEX Fall'90であった．

（清成啓次）

NEWS & NEW PRODUCTS

3 ボーランドジャパン，PC-9801版TURBO C++コンパイラを発売

（株）ボーランドジャパン（TEL 03-486-1400）は，PC-9800シリーズ対応の「TURBO C++」，デバッガ，プロファイラ，アセンブラが付属した「TURBO C++ PROFESSIONAL」を発売した．C++は，90年代の主流といわれるオブジェクト指向プログラミングを自然な形でC言語に取り入れた言語で，TURBO C++は，「TURBOシリーズ」の速度と使いやすさを，C++ネイティブコードコンパイラ（AT&T C++ Ver.2.0に準拠）として実現した．新しく改良された統合環境「Programmer's Platform」，640KBのメモリの壁を破る「VROOMM」（Virtual Runtime Object-Oriented Memory Manager）オーバーレイマネージャを装備している．

Programmer's Platformは，プログラムの編集，コンパイル，リンク，デバッグの作業をすべてマウスを使って行えるオーバーラップウィンドウ環境，Turbo AssemblerやTurbo Debuggerなどの外部プログラムを簡単に呼び出すことのできるトランスファ機能などを備えている．

VROOMMオーバーレイマネージャは，オーバーレイスワップ先として，ディスク以外にEMSメモリやプロテクトメモリを使用することができ，動的にオーバーレイのバッファサイズを変更することができる．

また，同社では，発売記念キャンペーンとして，1991年3月末日までの期間，1万円引きのキャンペーン価格を設定している．

■おもな仕様

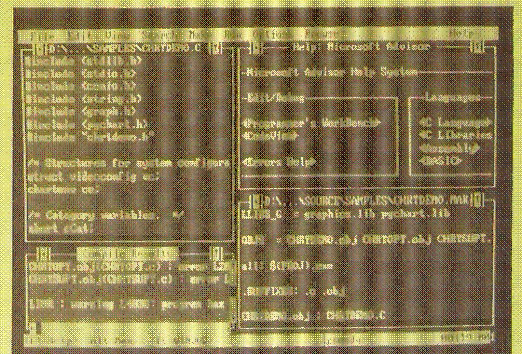
対応機種：PC-9800シリーズ（ハイレゾモード対応，LTをのぞく），EPSON PC-286/386シリーズ／環境：MS-DOS Ver.2.11以降，必要メモリ640KB以上，バスマウス対応，統合環境版の使用にはハードディスクが必要／価格：TURBO C++　39,800円（29,800円：1991年3月末日まで），TURBO C++ PROFESSIONAL　59,800円（49,800円：1991年3月末日まで）

4 マイクロソフト，MS-C Ver.6.0を発売

マイクロソフト（株）（第一営業部セールスインフォメーションセンター：TEL 03-363-1201）は，「Microsoft C Professional Development System Version 6.0」（日本語版）を発売した．これは，88年に発売されたバージョン5.1からのバージョンアップで，今回発売されたバージョン6.0では，プログラム開発の生産性を向上させる新しい開発環境「プログラマーズ・ワークベンチ」，大容量・高速実行を実現した「コード・ビュー」などを装備している．

新開発環境「プログラマーズ・ワークベンチ」は，定型作業を自動実行するプロジェクト指向性，コードを追加して機能を拡張することができるオープンアーキテクチャ，プログラムの構造をわかりやすく画面に表示する「ソース・ブラウザ」などを備えている．

「コードビュー」は，デバッガ自身のコードをEMSメモリに追い出す機能をサポートし，ユーザーインタ



ワンラインNEWS
丸紅エレクトロニクスは大容量の光ディスク搭載のサーバーを発売した．同機は従来の6，7倍の最大1テラバイト（1テラは1000ギガ）の容量をもつ．

NEWS & NEW PRODUCTS

ーフェースも全面改良された。

その他,「ベースド・ポインタ」や新しいプロシージャコール「_fastcall」のサポート,インラインアセンブラの統合化などの強化がなされた。

また,DLLとマルチスレッドのフルサポート,DLLとマルチスレッドプログラムの実行ファイルの作成を簡単にするオプション,デバッグ機能の改善など,OS/2に対応した機能をサポートしている。

■おもな仕様

対応機種:PC-9800シリーズ(ハイレゾリューションモードを除く80286,80386マシン)/環境:MS-DOS Ver.3.1以降,必要メモリ640KB以上,OS/2 Ver.1.1以降,必要メモリ1MB以上,6MB以上の空き領域のあるハードディスク/価格:98,000円(バージョン5.1からのバージョンアップ:25,000円,バージョン4.0以前からのバージョンアップ:40,000円)

5 ライフボート,C言語用画面作成ツールライブラリを発売

(株)ライフボート(TEL 03-293-4711)は,MS-C 5.1/6.0,Lattice C用の画面作成ツールライブラリ「PPII」を発売した。ポップアップメニュー,仮想画面によるスクロール・ウィンドウ,オーバーラップウィンドウなど多彩な機能が利用でき,マウスにも対応している。価格は,MS-C,Lattice C用とも98,000円。

6 吉正電子,MS-C 6.0およびQuick-C 2.0対応の開発支援ライブラリを発売

吉正電子(株)(TEL 03-5371-3041)は,MS-C 6.0およびQuick-C 2.0のオンラインヘルプに対応したアプリケーション開発支援用ライブラリ「Graphic Window Library」「ISAM Library」を発売した。価格は,両製品とも23,000円。

7 富士通,FM TOWNSの新機種を発売

富士通(株)(TEL 03-5690-0032)は,内蔵ハードディスクの大容量化,5万ゲートの超LSIの採用による高集積化により,従来機種より約20%の低価格を実現した,「FM TOWNS」の新機種を発売した。

発売された2機種4タイプは,2MBのメインメモリを標準装備し,通信ソフト,フルスクリーンエディタなど,FM TOWNSユーザーの開発した5種のフリーウェアが組込まれたシステムソフトウェアが標準添付されている。

また同時に,既存ユーザー向けに,新内蔵拡張スロットを採用した新I/O拡張ユニットの提供,TOWNSシステムソフトウェアのバージョンアップサービス,モデル1/2ユーザーを対象に,CD-ROM制御用マイコンの交換サービスを行う。

■おもな仕様

プロセッサ:80386(16MHz)/メモリ:2MB/外部記憶装置:CD-ROMドライブ×1・3.5インチFDD×1(モデル10F),CD-ROMドライブ×1・3.5インチFDD×2(モデル20F),CD-ROMドライブ×1・3.5インチFDD×2・40MB HDD×1(モデル40H),CD-ROMドライブ×1・3.5インチFDD×2・85MB HDD×1(モデル80H)/拡張スロット:2/価格:298,000円(モデル10F),323,000円(モデル20F),473,000円(モデル40H),623,000円(モデル80H)



NEWS & NEW PRODUCTS

8 日本IBM，PS/55の新モデル4機種ほかを発売

日本アイ・ビー・エム(株) (TEL 03-586-1111) は，80486を搭載した業界最高速モデルを含むPS/55の新モデル4機種を発売した．

新モデルは，CPUに32ビット80486 (33MHz) を搭載し，PS/55シリーズの最上位機種に位置づけられる「PS/55モデル5580-Y」(写真)．80486 (25MHz) を搭載し，現行のモデル5550-Vに比べ商用計算で約1.8倍，数値演算で約3倍の処理能力を実現した高機能デスクトップモデル「PS/55モデル5560-W」．80386(20MHz) を搭載し，コンパクト設計，ハイコストパフォーマンスを実現した「PS/55モデル5540-T」．80386SX (16MHz) を搭載し，通信機能を充実したラップトップモデル「PS/55モデル5535-S」の4機種．

また同時に，再生紙にも対応したレーザプリンタ「IBM5587-H01」(758,000円) も発売された．

■モデル5580

プロセッサ：80486 (33MHz)，8KBキャッシュメモリ／メモリ：4MB／外部記憶装置：3.5インチFDD×1・160MB HDD (5580-Y0B)，3.5インチFDD×1・320MB HDD (5580-Y0C) ／拡張スロット：32/16ビット用フルサイズ×6／価格：2,600,000円 (5580-Y0B)，3,000,000円 (5580-Y0C)

■モデル5560

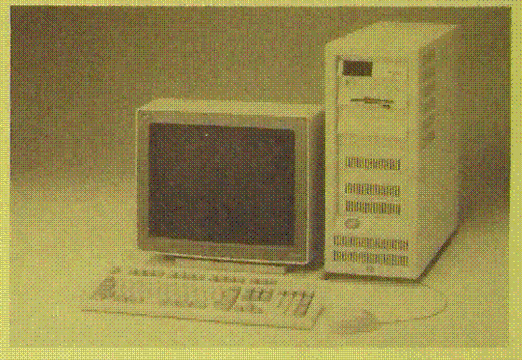
プロセッサ：80486 (25MHz)，8KBキャッシュメモリ／メモリ：4MB／外部記憶装置：3.5インチFDD×1・80MB HDD (5561-W08)，3.5インチFDD×1・160MB HDD (5561-W0B)，3.5インチFDD×1・320MB HDD (5561-W0C) ／拡張スロット：32/16ビット用フルサイズ×3，ハーフサイズ×1／価格：2,000,000円 (5561-W08)，2,200,000円 (5561-W0B)，2,600,000円 (5561-W0C)

■モデル5540

プロセッサ：80386 (20MHz) ／メモリ：2MB (5541-T04)，4MB(5541-T08)／外部記憶装置：3.5インチ FDD×1・40MB HDD (5541-T04)，3.5インチFDD×1・80MB HDD (5541-T08) ／拡張スロット：32/16ビット用フルサイズ×2／価格：768,000円 (5541-T04)，998,000円 (5541-T08)

■モデル5535-S

プロセッサ：80386SX (16MHz) ／メモリ：2MB／外部記憶装置：3.5インチFDD×1・40MB HDD／拡張スロット：16ビット用フルサイズ×1／ディスプレイ：16階調，バックライト付／価格：607,000円 (5535-S04)，635,000円 (5535-SJ4，IBM DOSバージョンJ4.0/V付属)



9 富士通，ノートブック型パソコンFM R-CARDを発売

富士通(株) (FMインフォメーションセンターTEL 03-646-0816) は，薄さ26.5mm，単三乾電池2本で8時間稼働する，重さ1kgを切ったノートブック型パソコン「FM R-CARD」を発売した．

外部記憶装置には，業界標準のICメモリカード (JEIDA Ver.4仕様準拠) を採用し，従来のフロッピーディスクに比べ，約1/1000の省電力と約100倍のアクセススピードを実現した．ICメモリカードは，フロッ

ワンラインNEWS
NTTはISDNのDSU (回線接続装置) をレンタルだけでなく買取もできるようにした．同時に高速デジタル伝送サービスについても一部の使用料を値下げした．

NEWS & NEW PRODUCTS

ピーディスクと同様にデータの読み書きが行えるRAMタイプの「ICメモリカード(SRAM)-128KB/256KB/512KB/1MB」と，低価格での供給が可能なROMタイプのICメモリカードがあり，アプリケーションはROMタイプのICメモリカードで提供される．ほかのFM Rシリーズとのデータ互換は，「ICメモリカードリーダライタ」(180,000円)で行うことができる．本体ROMには，MS-DOS Ver.3.22，予定表，通信機能などをもつ「NBツール」，アプリケーションなどの起動を行う「NBメニュー」などの「NBソフトシリーズ」を実装している．また同時に，本体とプラグイン接続が可能なモデム(36,000円)，I/O拡張ユニットなどの周辺機器も発売された．JISキーボードタイプと親指シフトキーボードの2モデルがある．

■おもな仕様

プロセッサ：80C286(8/4MHz)／メモリ：640KB／外部記憶装置：ICメモリカード×2／拡張スロット：1／ディスプレイ：2階調反射型／価格：238,000円(128KB ICメモリカード1枚標準添付)



10 アップルコンピュータジャパン，Macintoshの新モデルを発売

アップルコンピュータジャパン(株)(TEL 03-224-7000)は，3機種のMacintoshの新製品を発売した．発売されたのは，従来のSEと同位機種で，68000を搭載した低価格機「Macintosh Classic」，薄型でコンパクトな設計，最大256色のカラーをサポートした「Macintosh LC」(写真)，68030(20MHz)を搭載し，最大256色のカラー，256階調のグレイスケールをサポートした「Macintosh IIsi」の3機種．Macintosh IIsi，LCは，サウンド入力機能を備え，声を録音・保存できるマイクロフォンを内蔵．また，256色カラーをサポートした12インチRGBモニター(100,000円)，12インチモノクロモニター(68,000円)も発売した．

■Macintosh Classic

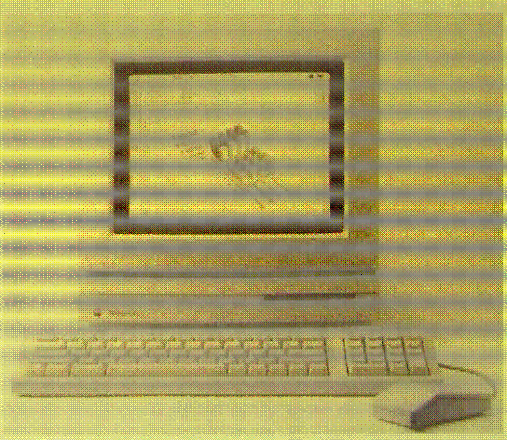
プロセッサ：68000(8MHz)／メモリ：2MB／外部記憶装置：1.44MB FDD×1(FDモデル)，1.44MB FDD×1・40MB HDD(HDモデル)／ディスプレイ：9インチモノクロCRT内蔵／価格：198,000円(FDモデル)，298,000円(HDモデル)

■Macintosh LC

プロセッサ：68020(16MHz)／メモリ：2MB／外部記憶装置：1.4MB FDD×1(FDモデル)，1.4MB FDD×1・40MB HDD(HDモデル)／価格：318,000円(FDモデル)，418,000円(HDモデル)

■Macintosh IIsi

プロセッサ：68030(20MHz)／メモリ：2MB，5MB／外部記憶装置：1.4MB FDD×1・40MB HDD(2MB/HD40モデル)，1.4MB FDD×1・80MB HDD(5MB/HD80モデル)／価格：708,000円(2MB/HD40モデル)，858,000円(5MB/HD80モデル)



NEWS & NEW PRODUCTS

11 日本電気，A5ファイルサイズのハンディパソコンを発表

日本電気（株），日本電気ホームエレクトロニクス（株）（パソコンインフォメーションセンターTEL 03-452-8000）は，PC-9800シリーズと互換のA5ファイルサイズのハンディパソコン「PC-98HA」（HANDY98）を発表した．統合ソフトMS-Worksをメモリカードで標準装備し，バッテリ駆動7時間．また，3.5インチフロッピーディスクドライブ，内蔵モデムスロットを装備した別売のドッキングステーション（50,000円，PC-98HA-01）を接続すれば，PC-98LT用のソフトを使うことができる．本体ROMには，MS-DOS Ver.3.1，N88-BASIC，6万語の辞書ROM，スケジューラなどを実装している．本体カラーは，ブラック，ホワイト，レッドの3色がある．

■おもな仕様

プロセッサ：V50（10MHz）／メモリ：640KB／外部記憶装置：RAMドライブ（1.25MB）×1／拡張スロット：1／ディスプレイ：2階調反射型／価格：198,000円



12 エプソン，32ビットマシン4機種を発表

セイコーエプソン(株)，エプソン販売(株)（エプソンPCインフォメーションセンターTEL 03-377-3531）は，ハイレゾリューションモード搭載の32ビットデスクトップパソコン「PC-386G」とPC-386Vの後継機にあたる「PC-386S」，NTN型カラー液晶ディスプレイを搭載した32ビットラップトップ「PC-386LSX」，着脱可能なハードディスクパックを採用し，RAMドライブカードの装着も可能なノートタイプパソコン「PC-386NOTE A」(写真)の4機種を発表した．

■PC-386G

プロセッサ：80386（33MHz）／メモリ：1.5～1.6MB／外部記憶装置：5インチFDD×1（F1），5インチFDD×2（STD），5インチFDD×2・100MB SCSI対応HDD（H100）／拡張スロット×2（内部）／価格：725,000円（F1），750,000円（STD），950,000円（H100）

■PC-386S

プロセッサ：80386（25MHz）／メモリ：1.6MB／外部記憶装置：5インチFDD×1（F1），5インチFDD×2（STD），5インチFDD×2・40MB HDD（H40）／拡張スロット×2（内部）／価格：473,000円（F1），498,000円（STD），638,000円（H40）

■PC-386LSX

プロセッサ：80386SX（16MHz）／メモリ：640KB／外部記憶装置：3.5インチFDD×2・40MB HDD



NEWS & NEW PRODUCTS

／拡張スロット×2／ディスプレイ：8・16色NTN型LCDカラー液晶／価格：800,000円

■PC-386NOTE A

プロセッサ：80386SX（16MHz）／メモリ：640KB／外部記憶装置：3.5インチFDD×1・ICカードドライブ×1／ディスプレイ：8階調FTN型サイドライト付LCD／価格：268,000円，298,000円（RAMドライブカード装着），398,000円（ハードディスクパック装着）

13 三菱電機，32ビットノート型AXマシンを発表

三菱電機(株)（TEL 03-5487-4840）は，同社のAXパーソナルコンピュータ「MAXY」シリーズに32ビット80386SXを搭載した，ノート・ブック型パソコン「MAXYNOTE386」を発表した．MAXYNOTE386は，バッテリ駆動2時間，ワンタッチで交換が可能な超小型2.5インチハードディスクを内蔵し，ハードディスクには，MS-DOS，MS-Windows，MAXYメニューシステム，ワープロ・表計算などの機能をもつ統合ソフト「MAXYMATE」がインストールされている．ディスクアクセスやキーボードのモード設定，パワー状況などをアイコンで表示する液晶システムパネル，レジューム機能などもサポートしている．

■おもな仕様

プロセッサ：80386SX（16MHz）／メモリ：1MB／ディスプレイ：バックライト型TST液晶8階調／外部記憶装置：1.44MB 3.5'FDD×1・20MB HDD×1（M3122A-12（M）），40MB HDD×1（M3122-A14（M））／拡張スロット：1（MAXYNOTE386専用サイズカード）／価格：478,000円（M3122-A12），418,000円（M3122-A12M＊），578,000円（M3122-A14），518,000円（M3122-A14M＊）＊M3122-A12M，M3122-A14Mはディスプレイ別売タイプ



14 東芝，DynaBookの新機種を発売

(株)東芝（TEL 03-457-2930）は，CPUに80C286を搭載し，現行モデルの約2倍の処理速度を実現した「DynaBook286 J-3100GS001モデル」を発売した．電源部の小型化などによる軽量化，新型FLサイドライト付き液晶ディスプレイの採用により従来機に比べ，約3.5倍の明るさを実現，オプションのバッテリパックを装着することにより，最大5時間まで使用できる．

■おもな仕様

プロセッサ：80C286（12MHz）／メモリ：1.5MB／外部記憶装置：3.5インチFDD×1／ディスプレイ：FLサイドライト付液晶／拡張スロット×1／価格：198,000円



NEWS & NEW PRODUCTS

15 ASTリサーチ・ジャパン，PC/AT・PC-9801互換マシンを発表

ASTリサーチ・ジャパン（株）（TEL 03-818-0710）は，80386SXを搭載し，IBM PC/ATとNEC PC-9801に高い互換性をもつ，32ビットマシン「DualStation 386SX/16」を発表した．システムディスクを挿入すると，AT／98どちらのモードかを自動認識する．販売は，スワイヤトランステック(株)，大倉商事(株)，プロサイド（株)の3社の販売代理店を通じて行われる．

■おもな仕様

プロセッサ：80386SX（16MHz）／メモリ：1MB／外部記憶装置：5インチFDD×2（モデル5），5インチFDD×2・40MB HDD（モデル45），5インチFDD×2・110MB HDD（モデル115）／拡張スロット：8／価格：389,000円(モデル5)，499,000円(モデル45)，569,000円（モデル115）



16 ソニー，RISC NEWSシリーズ最上位機を発表

ソニー(株)（TEL 03-448-3553)は，ワークステーション「NEWSシリーズ」の最上位機として，25MIPSの処理速度を備えた「NWS-3870」を発表した．同機はデスクトップ型ワークステーションとしては最大容量1.25Gバイトのハードディスクと64Mバイトのメインメモリを内蔵している．価格は，5,750,000円．

17 東芝，40MHzの32ビットRISCプロセッサを搭載したWSを発売

(株)東芝（TEL 03-457-2100）は，デスクトップ型エンジニアリングワークステーション「SPARC」シリーズに，RISCプロセッサとグラフィック加速器を搭載した「AS4075GX」，「AS4075GS」，「AS4075GT」を追加発売した．「AS4075GS」と「AS4075GT」は，24ビットフルカラー（1670万色）機能を備えている．価格は，3,630,000円から．

18 富士通，「Sファミリーエンジニアリングワークステーション」ラインナップ強化

富士通(株)（TEL 03-261-3211)は，デスクトップ型エンジニアリングワークステーション「Sファミリー」シリーズに，RISCプロセッサとグラフィック加速器を搭載した「S-4/2 GX」，「S-4/2 GS」，「S-4/2 GT」を発売した．「GS」と「GT」は，24ビットフルカラー（1670万色）機能を備え，3次元の高速表示が可能．価格は，2,750,000円から．

19 コンテック，PC-9801N/NS用のI/Oユニットを開発

(株)コンテック（TEL 03-769-1061）は，NECのノートパソコン（PC-9801N/NS）用のI/Oユニット「NOTE-PAC(98)-2A」を開発した．98用の拡張ボードが2枚まで実装できる．また，外付けフロッピーディスクドライブを直接接続できる．価格は78,000円．

ワンラインNEWS
日本貨物運送協同組合会連合と全日本トラック協会は共同でパソコン通信による貨物情報システムを開発した．パソコンの端末から求車情報などを得られる．

NEWS & NEW PRODUCTS

20 エプソン，日本語ページプリンタの新機種を発売

セイコーエプソン（株）とエプソン販売（株）（TEL 03-377-7001）は，日本語ページプリンタ「LP-7000G」，「LP-3000」を発売した．解像度はLP-7000Gが240dpi，LP-3000は300dpi．両機種ともアウトラインフォントを搭載している．価格は，LP-7000Gが471,000円，LP-3000が254,000円．

21 アイ・オー・データ機器，All in Note用増設メモリボード発売

（株）アイ・オー・データ機器（TEL 0762-21-2373）は，シャープのAll in Noteおよび三菱MAXY NOTE用1MB増設メモリボード「SH-AN34-1M」を発売した．拡張メモリ，EMSメモリ，画面退避メモリとしても利用でき，最大2枚まで増設可能．価格は，30,000円．

22 日本IBM，IBM DOSバージョンJ4.0/Vを発売

日本IBM（株）（TEL 03-586-1111）は，新機種の32ビットラップトップ「PS/55モデル5535-S」や低価格パーソナルコンピュータ用のオペレーティングシステム「IBM DOSバージョンJ4.0/V」を発売した．米国でPS/2の標準表示システムとなっている“VGA”（640×480ドット）の上で日本語環境を実現するために開発されたもので，16ドット文字で日本語を表示することができる．新製品のPS/55モデル5535-S，5550-VおよびPS/55Zで利用することができる．解像度以外の機能は，IBM DOSバージョンJ4.0と同等．価格は，40,000円．

23 ライフボート，DPMI委員会に参加

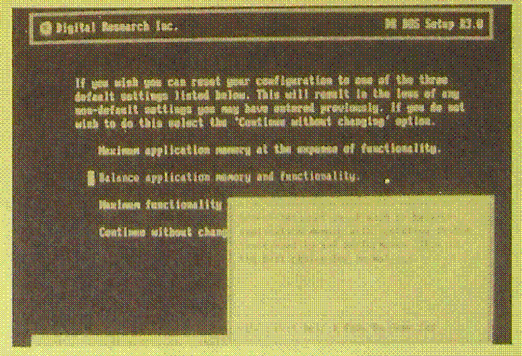
（株）ライフボート（TEL 03-293-4711）は，米国のDPMI（DOS Protected Mode Interface）委員会にアドバイザリ・メンバーとして参加することが決定．DPMIは，MS-DOSで286や386のCPUで1Mバイトを超えるメモリの管理の規格．日本からのこの委員会への参加は同社が初めて．

同委員会には，米マイクロソフト，インテル，ボーランド，ロータス・デベロプメント，フェニックス・テクノロジー，クォーターデッキ，ファー・ラップ社などが加入している．

DPMIは，Windows 3.0のメモリ管理にとり入れられて話題を呼んでいる．

24 デジタル・リサーチ・ジャパン，日本語DR-DOS 5.0を販売開始

（株）デジタル・リサーチ・ジャパン（TEL 03-490-1551）は，MS-DOS互換OSである日本語DR-DOS 5.0の販売を開始した．DR-DOSはROM化を考慮し，パソコンから組込みコンピュータまで幅広い対応をしている．また，80286以上のCPUではOSのカーネル部分やTSRプログラムをアッパーメモリに移動するため，フリーエリアを確保することが可能となっている．ほかにもバッテリ消費を大幅に延長する機能も備わっている．



◆◆◆◆ 佐々木哲の ◆◆◆◆

N.Y. システム通信

SYSTEM N Y NEWS

コンピュータ産業はますます巨大化，複雑化しています．「古き良き時代」はさり，もうアイデアだけのビジネスは成り立たなくなりつつあります．これからはマーケットのニーズをとらえ，いかにシステムを作るかがより重要になっていきます．

「マニア」「職人」の時代は終わりつつあるのでしょうか．この視点からみると，旧態依然で国内でしか通用しないものを作り続ける日本には不安を感じます．輸出しすぎで各国からたたかれる日本の産業のなかにあって，ソフトウェア産業だけは大幅な輸入超過が続きます．

まだまだ活気を失わないここアメリカの現地から，PC関連を中心に各種製品，市場動向をピックアップしていきたいと思います．

Windows 3.0，DOS 5.0，OS/2 2.0

90年5月22日にMS-Windows 3.0が発表されたときは，全米9か所を衛星中継でつないで盛大なショーが行われ，1万人の聴衆が参加しました．もちろん，みんな実物のWindows 3.0をおみやげにもらって帰りました．朝のABC TVのモーニングショーにまでビル・ゲイツが出演するという騒ぎでした．

以前のWindowsソフトウェアと互換性に欠けるWindows 3.0．驚くべきことは，このときすでに多くのソフトウェアハウスはWindows 3.0への対応をすませていたのです．半年以上前から，かなり広い範囲でWindows 3.0のベータ版が配布されていました．決してAldusのようにメジャーなところだけではありません．数人でやっているSharewareの小さなソフトハウスでさえもです．

近々発表される80386専用のOS/2 2.0もかなり長期間にわたってベータ版の配布が進んでいます．また，DOS 5.0が発表されようとしています．これももう広い範囲でベータテストが進んでいます．ハードウェアメーカーやソフトハウスだけでなく，企業，一般のユーザーグループまでも入手しています．本来のベータテストとしてフィールドでのバグの発見だけでなく，ソフトウェア，ユーザーの対応も行われていくのです．

マイクロソフトにはベータ版のコーディネータがいて，ここに申し込みさえすれば最新のものが手に入ります．ただし，OS/2 2.0は2,800ドルという高価なものです．DOS 5.0ではベータ版を必要とする十分な理由をレターで送らなければなりません．

より多岐に複雑になっていくソフトウェア．発表されてからでは対応も遅れます．ましてや日本の機種に移植されるのを待っていてはいつになることやら．一度IBM互換機を手にいれて試してください．

Windows Slueth，Mac Slueth

エンドユーザーに使いやすいはずのWindows 3.0．各種メモリ環境，LAN，ディスクコントローラなどハードウェアの構成によりいったんトラブルが発生すると，原因を見つけるのが困難です．そこに強力な味方が出てきました．

PCの診断ソフトで知られるSystem Slueth（探偵）の発売元Dariana Technology Group*1（米国カリフォルニア）では，新たにWindows 3.0バージョンとMacintoshバージョンを発売しました．各種のハードウェア，ソフトウェアの組み合わせで発生するトラブルを解析し，報告，修正してくれます．オフィスのPCの面倒を見る人には強い味方です．

＊1:Dariana Technology Group Phone: 714-994-7400

VGAと消えたハーキュリーズ

1987年4月PS/2とともに紹介されたVGAディスプレイ標準（640×480ドットで16色，320×200ドットで256色）ですが，その後互換カードも広がり，値段も安くなってきています．

当初500ドル以上していたVGAカード，ここのところ積極的な動きを見せるATI Technology*2（カナダ）はとうとう定価99ドルのVGAカードを発売しました．もうVGA以下のものを使う理由はありません．ほかのメーカーも実売価格では100ドル台でりっぱなものが手に入ります．

最近，雑誌の広告を見ていて気がつくのですが，ハーキュリーズグラフィックカード（720×348ドット，モ

ノクロ)がすっかり姿を消しました．1981年に紹介されてから，たとえCPU本体が486であろうと安価なシステムをくむために便利だった貧乏人の友達Herculesカードよ，安らかに．

＊2:ATI Technologies 3761 Victoria Park Ave, Scarborough, Ontario, Canada M1W 3S2 Phone:(416)756-0718 Fax(416)756-0720

Super VGAとその環境

そのVGA規格ですが，本家IBMのものは解像度最高640×480ドットで16色，あるいは320×200ドットで同時256色です（日本の代表機種PC-9801では640×400ドット，16色）．VRAMとしては256KB載っています．

VGAはその仕様上，比較的簡単に拡張が可能です．互換VGAボードでは純正モードに加えて，さらに高解像度を持たせているものが数多くあります．これらはSuper VGAと呼ばれ，800×600ドット，1024×768ドットなどの高解像度，あるいは640×480ドットで同時256色など，より多くの色をサポートしています．IBM自身でさえ，XGAの名の元にVGAを拡張する動きもあると聞きます．

Super VGAでは，標準の倍の512KBのメモリが装備されたものが主流です．代表的なものはVideo-7 VRAMなどです．このメモリ量では，800×600ドットで256色か1024×768ドットで16色までサポートされます．

VGAのように描画CPUの載っていないダムなグラフィックアダプタでは，これだけの巨大なVRAMは本体のCPUへの負荷が問題です．一般にEGA，VGAなどのVRAMには数10サイクルのかなりなウェイトがかかっています．いくら高速なCPUでも，グラフィック中心のソフトを走らせると遅いこと遅いこと．

この夏ごろから，さらに新しい世代のSuper VGAカードが市場に出てきました．メモリを1MBまで搭載可能．1024×768ドットで256色表示できます．しかもありがたいことに，メモリアクセス速度が以前のものと比べて10倍ほどと，かなり改善されています．それでいて今までのVGAと変わらぬ価格です．

Super VGAのもうひとつの問題は互換性です．これらの互換VGAは拡張方法が各社さまざまです．通常のVGAだけをサポートしたソフトではこれらの高機能の持ち腐れです．ところが，MS-Windowsの上で動くものなら大丈夫．Windows側がそのVGAカードをサポートすればソフト側では気にすることはありません．OS/2なども同様です．Super VGAカードではWindowsを含む代表的なソフトの表示ドライバがいっしょに提供されるのが普通です（ここは各メーカーのサポートの差が出るところでもあります）．

この非互換性の問題を業界で解決しようとする動きもあります．VESA[*3](Video Electronics Standards Association)規格と呼ばれ，ハードウェア，BIOSなどに一定の規格を設けようというものです．また，既存のSuper VGAを直接プログラムするためのデータを集めるにはCompuserveのGraphic Supportフォーラムが最適です．ここには今までの集大成であるVGAKITがPDSとして公開されています．

＊3:Video Electronics Standards Association 1330 South Bascom Avenue, Suite D San Jose, CA 95128-4502 Phone:(408)971-7525 Fax:(408)286-8988

今月の何だこりゃ？

コンピュータはユーザーもメーカーも人より速く動かしたいのが人情．かといって高速で動作させるには発熱が最大の問題です．Velox[*4] なるベンチャーメーカーがICECapなるものをOEM向けに発表しました．

半導体素子を使ったチップ用ミニチュアの冷却器．これをCPUの上に張り付けて発熱を抑え，CPUを規格を越えて動作させようというもの．とくにCPUまわりがひとつにまとまった80486には最適で，25MHz規格のものを50MHzで動作させることも可能とか．別に680x0でもRISCでもかまいません．なんでも，EverexやCompacなどの高性能互換機メーカーでは高速CPUが発表される前の試作に使っているとか．

＊4:Velox 2334C Walsh Ave., Santa Clara, CA 95051 Phone:(408)727-6100

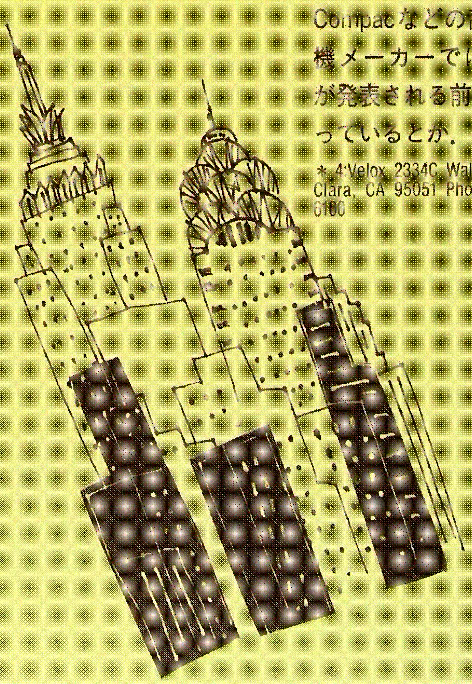
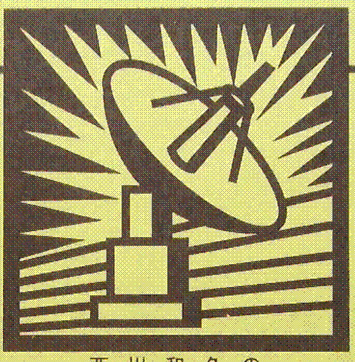


Illustration:MASAMI NOZUE

SYSTEM N Y NEWS

Illustration:NOBUKI FUKUHARA



西川和久の

希望的観測

国際化，標準化への新たな流れとなるか VGAで漢字を実現するDOSが登場

1 独自のハードウェアなしに日本語を実現

11月5日，かねてから噂のあったVGA上で日本語環境を実現するDOS『IBM DOS J4.0/V』が発売された．DOSに限った話ではないのだが，これまでは日本語処理をする場合，世界的視野から見ると全く互換性がない国産機で処理を行う必要があった．これは，国際性を問われている現在の状況にあって，日本にとってあまりにも不利な状況であった．

J-3100，PS/55，AXなどは，確かにIBM PCを日本語化したマシンではあるが，日本語処理用のハードウェアを独自に搭載しているため，ピュアなPCとは呼べない．日本語化のための特殊なハードウェアを必要とせず，全てをソフトウェアだけで処理してしまうこともできるこのDOSの持つ意味は非常に大きく，これまでの常識を根底から覆してしまうものである．PC上で日本語処理を可能にするソフトにEW＋があるが，これはワープロソフトという限られた枠内の話であり，グローバルな環境として扱えるものではなかった．

たとえばこれまでなら，海外から日本にきているビジネスマンなどは，自分が使うアプリケーションやデータをフロッピーで持ってきて，英語モードを持つ国産マシンでモードを切り替えて使用すればことは足りたのだが，逆に日本人が海外で日本語処理したい場合はマシンを持参して行かなければならなかった．

これでは海外出張のたびに大荷物でたいへんだし，手続き上非常に煩雑な処理が発生する．また海外では，同性能程度のマシンが半値近くで売られているが，買って帰っても日本語処理ができなければ話にならない．しかし，このDOSの発表によりフロッピーひとつで国内外問わず，日本語処理ができるようになったのだ．

2 「なぜもっと早く」と思うほど簡単なカラクリ

いまから考えれば，なぜもっと早くこのDOSが世の中に出てこなかったと思うほど簡単なカラクリであり，マッキントッシュを日本語化した状況と同じ考え方である．現在PCの世界ではCGA，EGAを経て，VGAがPCの標準的な画面モードとなっており，画面解像度は国内の代表的なマシンと同等の640×480ドットなのだ．これに16ドット漢字フォントをビットマップでスクリーン上へ展開するという最も単純なカラクリで構成され，漢字フォントデータはファイルから拡張RAMへ転送される．

J-3100もよく似たカラクリで日本語化を実現しているが，「画面モードAT&T B/W」「漢字フォントデータはROMで本体内蔵」という，あまり一般的ではないマシンに依存する手法で日本語化されているため，J-3100専用，すなわちローカルなものでしかなかった．しかし，この『IBM DOS J4.0/V』を使うためのシステムリクエストは次のような，ごく一般的なPCの構成であ

ワンラインNEWS
米シリコンバレーのベンチャー企業は日本企業による買収・投資に魅力を感じている．資金力のほか製造やマーケティングに強力な援助が期待できるため．

り，これならワールドワイドに見ても十分通用するはずだ．

VGA漢字DOSの動作条件

1. IBM PC/ATまたはPS/2互換機
2. CPU80286以上
3. メモリ1MB以上
4. VGA搭載

グラフィック画面に漢字フォントをビットマップ展開すると，「描画速度が遅いのでは!?」と懸念されるであろうが，CPU・ビデオ回路の高速化に伴いこの問題は解決されるはずだ．少しくらい速度を犠牲にしてもこのDOSの持つ意味は非常に大きいものだと思う．

3 少し見方を変えれば……

少し見方を変えれば，いろいろおもしろいことに気がつく．たとえば，日本に限らず，中国，韓国などDBCS（ダブル・バイト・キャラクター・セット）が必要な国にもフォントデータを変えれば適応可能なことは容易に想像がつく．「VGAの漢字DOS」というよりも，「VGA・DBCS・DOS」と呼んだほうがふさわしいかもしれない．またVGAの次世代画面モードとして噂されているXGA（解像度1024×768ドット）にこのDOSのカラクリを採用すれば，いとも簡単に24ドット漢字フォントが可能になる．DOSが出れば当然，「VGA漢字Windows」や「VGA漢字OS/2」なるものが登場する可能性も……などいろいろな想像をしてしまうのは私だけではないと思う．

4 メリットがあるのだから手を組めば……

個人的な希望になるが，このDOSをベースにJ-3100やAX陣営も手を結んでくれれば，PC-9801とFMRに代表される「純国産マシン軍団」に十分対抗できる「VGA漢字軍団」が形勢されるであろう．J-3100の海外版ではすでにVGAは標準搭載されているため，「国内外の生産ラインを一本化する」という大きなメリットが産まれる．AXもVGAを拡張したJEGAシステムが近々発表されるとも聞いている．これならスペック的には問題はないと思われる．

何よりもいい点は，ユーザー側にある．少しマシンの構成が違うために同じソフトをそれぞれのマシン用に購入する必要がなくなり，ソフトウェア資産の相互間有効活用につながる．また，世界にも通用するこのDOSをプラットホームとしてアプリケーションが使用できるわけだ．ただ，これらの統合化は各メーカーのいろいろな思惑があるため，すぐにとは行かないだろうが，できることなら手を組んで，いまこそハードウェア，ソフトウェア，ユーザーの国際化に手助けをしていただきたいと切望する．



NEXT STAGE

10月9日から11日の3日間，マルチメディア国際会議フォーラム（代表：塚本慶一郎／（株）アスキー，事務局：（株）電通）／日本経済新聞社主催によるマルチメディア国際会議'90が，千葉県幕張メッセの日本コンベンションセンターで開催された．
1989年の第1回につづいて今回も国内外から著名な講師陣が招かれ，マルチメディアの将来・技術・市場などについての基調公演，パネル討論が行われた．

マルチメディア国際会議'90開催
Windows 3.0で勢いづくDOS陣営

◆米国からキーパーソンを招いて基調講演

第1日目には，オートデスク・フェローであるテッド・ネルソン氏が「21世紀へのメディア進化」と題してハイパーテキストの概念について語り，つづいて米国マイクロソフト社会長であるウイリアム・ゲーツ氏が「マルチメディア市場開拓に向けて」としてWindows3.0とCD-ROMを用いたデモンストレーションとともに公演を行った．また，米国IBM社副社長であるルーシー・フェルドスタッド氏は，映画「乱」を題材としたマルチメディア作品のデモンストレーションを通じて，マルチメディアの楽しさを参加者にアピールした．

2日目は，マルチメディア市場予測の権威であるジョナサン・シーボルト氏から「マルチメディア市場予測」について，ルーカスアーツ・エンターテイメント社副社長ステファン・アーノルド氏から「エンターテイメント産業とマルチメディア」について，そしてマルチメディアソフトで有名なマクロマインド社会長マーク・キャンター氏から「マルチメディアの標準」について公演が行われた．

最終日には，ハイパーカードの設計者として有名なアップルフェローのビル・アトキンソン氏より「テクノロジー&エデュケーション」に関して，米国インテル社副社長デビッド・ハウス氏から「画像処理技術とその展望」について，そしてNTTヒューマンインタフェース研究所の安田浩氏が「画像圧縮と通信プロトコル」について公演を行った．



◆現場の事例を提示してパネル討論

パネル討論は，「マルチメディアの知的財産権」「マルチメディア未来論」「マルチメディア・アプリケーションの展望」，「マルチメディア・ソフト製作ガイダンス」「バーチャル・リアリティ」「マルチメディア動画処理技術」「教育とマルチメディア」など，それぞれのテーマにそって行われた．

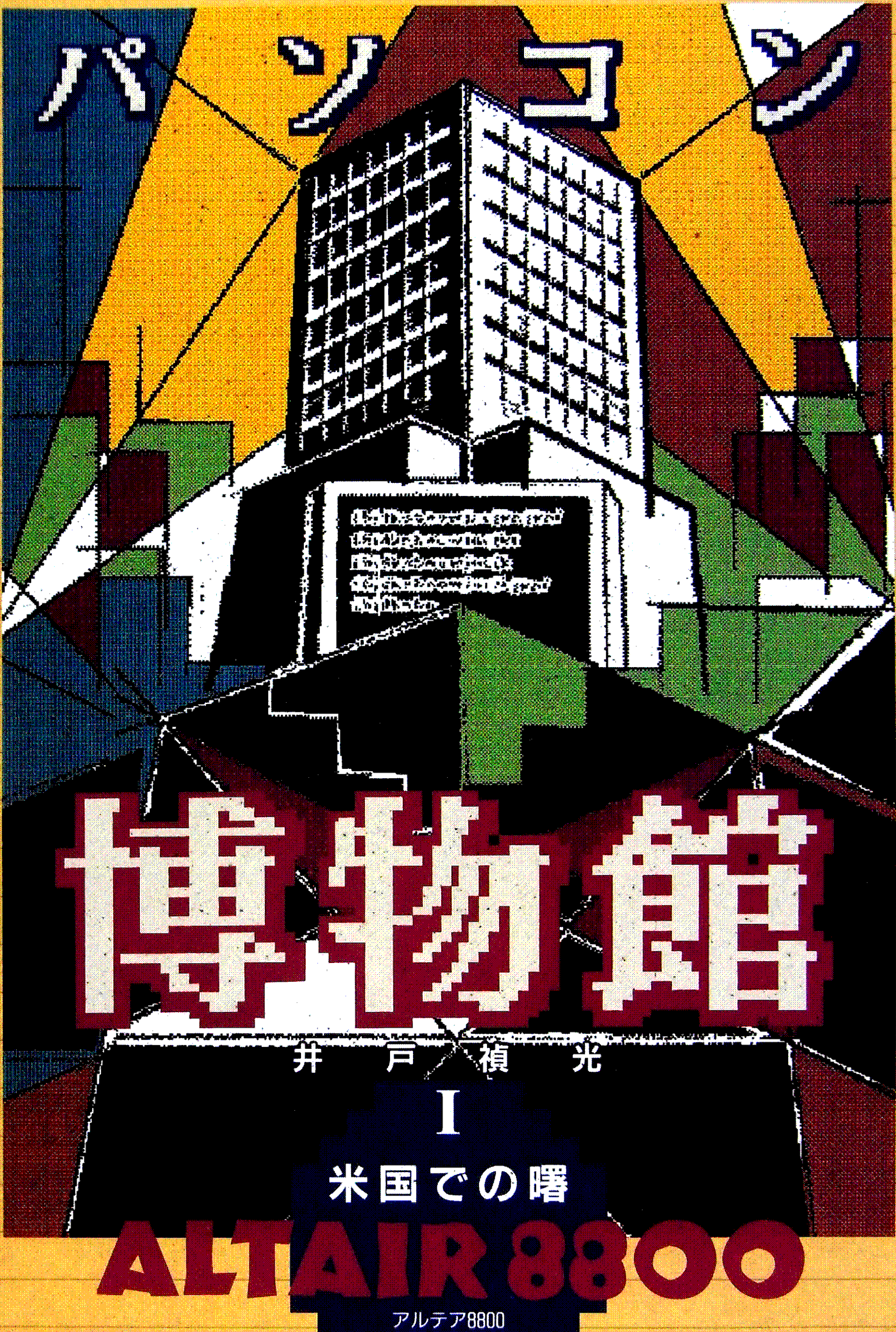
「マルチメディア・アプリケーションの展望」では，音楽家の加藤和彦氏がパネラーとして参加し，加藤氏が現在製作しているCD-ROMの紹介をしながらアーチストから見たマルチメディアについて論じた．また，「教育とマルチメディア」では，富山市立神明小学校教諭戸塚滝登氏から，マルチメディアを用いて小学生たちが木の年輪などから作曲を行っていく過程が紹介された．

◆開発関連はオーサリングソフトが主体

セミナー会場近くの展示ホールでは，関連企業26社によるマルチメディア製品の展示が行われたが，Macintoshと比較して，マルチメディアでは不利といわれているMS-DOSマシンが，Windows 3.0の発表によって勢いを盛り返しているのが印象的だった．

具体的な製品としては，CD-ROM・LD-ROM関連のハード・ソフトおよびマルチメディアを作成するためのオーサリングツールと呼ばれる一連のソフトが大部分を占めた．

ワンラインNEWS
NTTは年度末にも衛星デジタル通信サービスの回線使用効率を高め，収容できるユーザー数を拡大すると発表した．

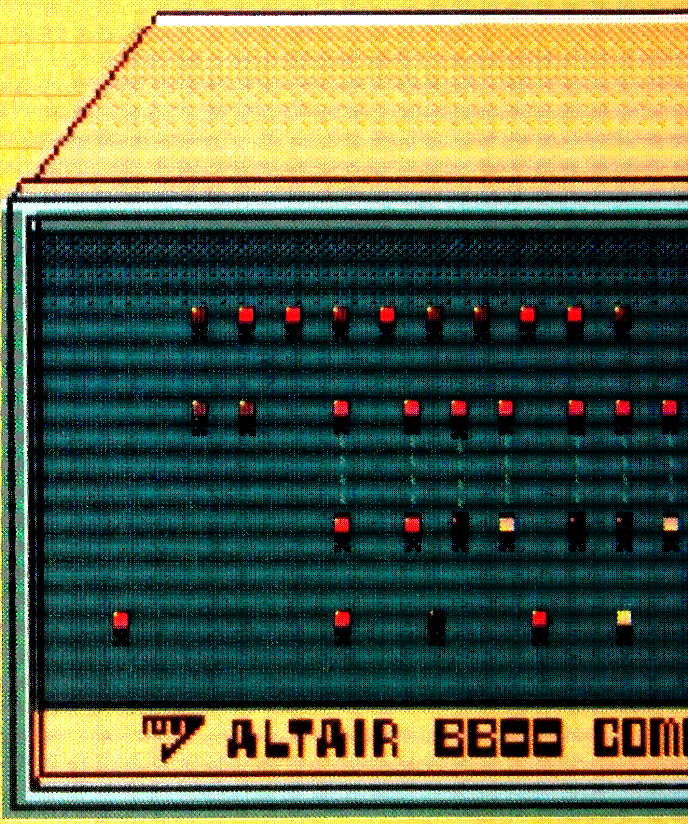


パソコンの歴史は，アルテア8800にはじまるといってよいであろう．1974年夏，ユニバック１にも匹敵するような機械が1000ドル以下で売り出されたのである．アルテア8800は，ニューメキシコ州アルバカーキ市にあったMITS社（Micro Instrumentation and Telemetry Systems）の製品であり，社長はエド・ロバート氏であった．

当初は，キットとして売り出され，ほかにも何社か売り出している会社もあったようである．しかし，通信販売をしたことと，ポピュラーエレクトロニクス誌の1975年１月号にのった記事のため，当時としては量販が行われ，社会的に大きな影響を与えた．キットの価格は439ドル，組立済みで621ドルであった．基本セットでは8Kのメモリターミナル，8K BASICを加えたもので1391ドル，強力なシステムとしては12Kのメモリとフロッピーディスクを加えたもので6650ドルであった．これは高いと思う人もいるかもしれないが，DECの１万ドルのシステムと同等のシステムを組むと5000ドルとなり，当時では安かった．８か月で4000台売り上げたが，この数字はアメリカのコンピュータの台数の２～３％に当たるといわれた．

個人でコンピュータをもつことにどれだけ意味があるかではなく，たんにステレオセットの半分の予算で，多少取扱いや利用面での不便があっても，コンピュータをもてるということは胸のわくわくすることであると考える人がいたのである．価格低下が可能になった理由は，インテル社が8080というICを売り出したからである．もとは電卓とかキャッシュレジスタなどへの利用を想定していたが，汎用コンピュータとしても十分使えることが証明されたのである．

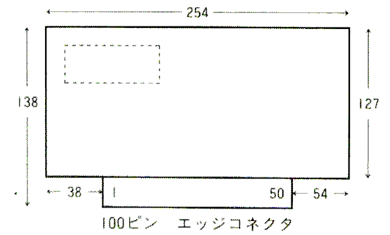
MITS社の功績は，汎用コンピュータとして必要なすべての部分（ターミナル，ディスク，プリンタ，インターフェース，それにサービス組織）を考慮していたことである．つぎつぎと現れる入出力装置のうち，コストパフォーマンスのよいものを自由につなぐためには，標準のコネクタを決め，信号条件を決めておくと便利である．MITS社では，S-100バスを開発した．技術情報がオープンであったため，いまでも多くのアマチュアが使っており，“ホビイスト標準バス”などともいわれている．このバスを採用したパソコンが次々と出た――たとえばIMSAI8080，POLY88，Cromemco, Northstar, Equinoxなど，さまざまであった．



S-100バスは，現在はZ80や6809のための改良を加えて，かつ16ビットのCPUにも対応できるようにしたIEEE696として定められている．しかし当時は，8080用であり，使えるメモリも1Kビットの2101Aしかなかった．そこでバスに8080用の信号線がそのまま出るようになっていた．メモリも入出力が分離していたので，S-100バスのデータ線は入出力が８ビットずつ分離して出ている．電源もスイッチングレギュレータがなかったので，各ボード上で安定化させるようになっていた．しかし，良い点としてはボード実装枚数が22枚と多く，中規模のシステムを作るには便利なことである．

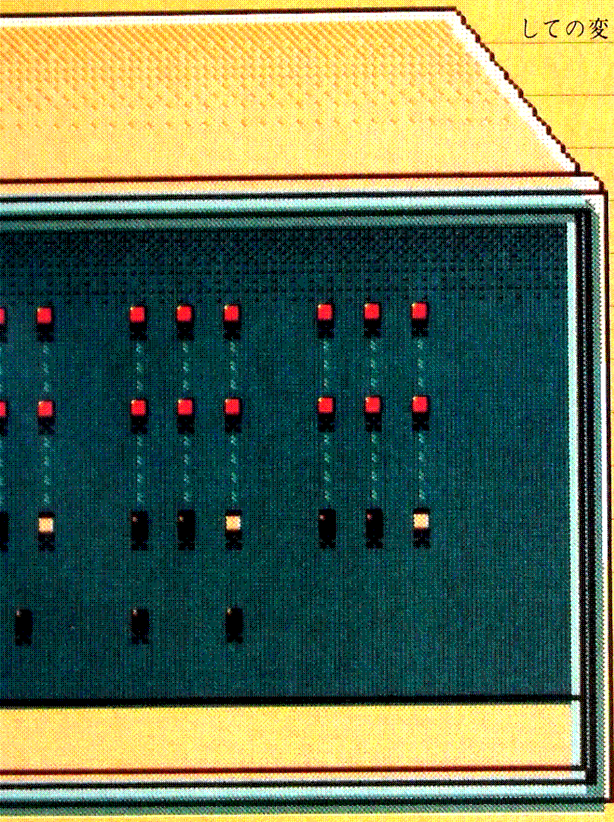
IEEE規格になって入出力線は，単方向だけでなく，“マスタが要求－SIXTRQ”，“スレーブが応える－SIXTN”の２本の線を使って16ビットの双方向信号を

S-100バス用のボードサイズ



左上の部分にレギュレータICを入れ放熱器をおくことが多い．

扱えるようになった．マスタは，SIXTRQを出してみて応答がなければ8ビット信号で片方向バスとして転送する．アドレス線も16本から24本となり，16MBまでアクセスできるようになった．この変更により転送速度は，12MHz×1バイトから6MHz×2バイトに変わるが，全体としての変



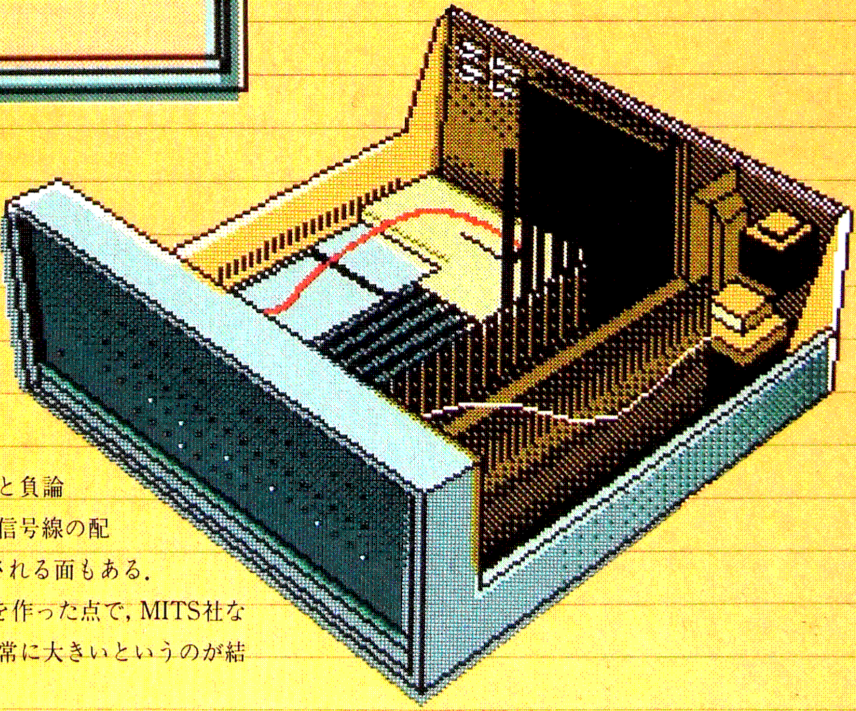
S-100バス主要信号表（オリジナル）

+8V　（+5Vの基）		1	51	+8V　（+5Vの基）	
+16V（+12Vの基）		2	52	−16V（−12Vの基）	
		3	53		
割込レベル	0	4	54	入出力機器をクリア	
〃	1	5	55	実時間クロック	
〃	2	6	56	8224ストローブ信号	
〃	3	7	57		
〃	4	8	58	レディ信号（フロントパネル）	
〃	5	9	59		
〃	6	10	60		
〃	7	11	61		
		12	62		
		13	63		
		14	64		
		15	65		
		16	66		
		17	67		
		18	68	メモリ書き込み可	
		19	69	プロテクト状態	
		20	70	メモリプロテクトする	
シングルステップ		21	71	RUNモード	
		22	72		
		23	73	8080への割込	
位相2クロック		24	74	8080へのHOLD	
位相1クロック		25	75	8080へのクリア	
ホールド		26	76	8080同期	
ウェイト		27	77	8080書き込み可	
割り込み可		28	78	8080DBIN	
アドレスビット	5	29	79	アドレスビット	0
〃	4	30	80	〃	1
〃	3	31	81	〃	2
〃	15	32	82	〃	6
〃	12	33	83	〃	7
〃	9	34	84	〃	8
出力データビット	1	35	85	〃	13
〃	0	36	86	〃	14
アドレスビット	10	37	87	〃	11
出力データビット	4	38	88	出力データビット	2
〃	5	39	89	〃	3
〃	6	40	90	〃	7
入力データビット	2	41	91	入力データビット	4
〃	3	42	92	〃	5
〃	7	43	93	〃	6
		44	94	〃	1
8080　OUT		45	95	〃	0
8080　IN		46	96	8080割り込み可状態	
8080　読みだし		47	97	8080書き込み出力状態	
8080　停止		48	98	8080スタック状態	
クロック（2MHz）		49	99	電源ON時のクリア	
アース		50	100	アース	

以上の信号表から信号のピンアサインメントがゴチャゴチャであることがわかると思う．

更はない．CPU（パーマネントマスタ）は1台のみで，DMAなどを行うテンポラリマスタは16台まで使える．テンポラリマスタは，CPUにバスの使用権を譲り受けてからデータを送り出すことになる．

一方，欠点としては正論理と負論理の信号がまざっていたり，信号線の配置が滅茶苦茶だったり，批判される面もある．しかし，パソコン発展の基礎を作った点で，MITS社ならびにS-100バスの功績は，非常に大きいというのが結論である．



特集 プログラマの仕事場

パソコンプログラマの開発環境と
ノウハウを一挙に公開

PWS
Programmer's Working Space

パソコンに先行して普及した汎用コンピュータによって，プログラマという言葉は多くの人が認識するところとなった．しかし，その時代のプログラマはコンピュータ業界にかかわりのある人でなければ，その姿を見ること

プログラマの生態学

コンピュータの登場とプログラマ社会の形成

現実のプログラマを観察する前にプログラマが如何にして発生し，共同社会を形成していったかを見ておく必要があるだろう．それは，コンピュータ機械の完成とその発展に大きく影響されている．人類は，算盤，歯車式計算機など洋の東西を問わず計算機械への長い努力の歴史をもっている．その歩みが急速な変化を見せたのは，今世紀になってからのことである．最初の電子計算機エニアックが完成したのは1946年のことであった．エニアックにはじまるコンピュータプログラマの発生，そしてその成熟と分化をまず見てみることにしよう．

コンピュータ帝国の誕生

エニアックは，17,500本の真空管からなる重さ30トンにおよぶモンスターであった．そこにおいては，配線をつなぎなおすことがすなわち機械に対する命令であり，計算を実行することであったと伝えられている．これを最も原始的なプログラミングであるとすれば，エニアックを作りだし，配線を変えて計算を行った研究者が，プログラマ族の最も古い祖先であると考えることができる．

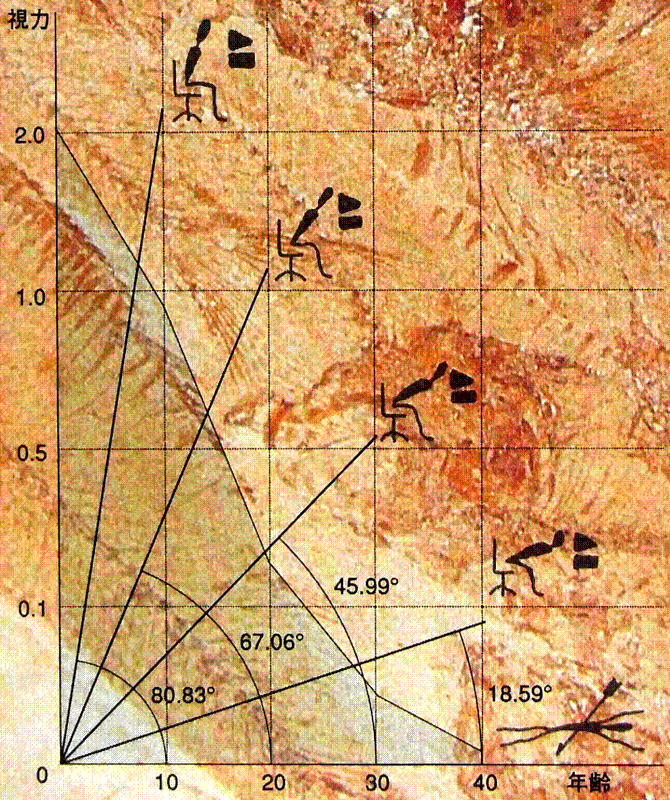
しかし，エニアックは商業ベースにのるようなものではなかった．コンピュータを商業的に

練習問題① 最初に商業用コンピュータを発売したのはどのメーカーか． ①バローズ ②ユニバック ③IBM

はできなかった．そうした状況を変えたのは「マイコンブーム」であった．マイコンがフツーの家庭に侵入し，若きハッカーたちが誕生することになった．中学生や高校生でハッカーになった少年少女（少女は少ないが）は，やがてパソコンを手にし，パソコン業界でプロとして活躍することになる．パソコン業界も発展分化をとげ，あまたのソフトハウスを誕生させたのである．
ソフトハウスやプログラマは存在するものの，これまでその生態が明らかにされることはなかった．「みんなMS-DOSでC言語を使っている」，確かにその通りなのだが，どのような環境を構築して，どういう仕事をしているのかは個人的なかかわりのなかでしか知ることができなかった．そこで，パソコンプログラマの開発環境を白日のもとにさらし，パソコンプログラマの世界を見せてしまおうというのが，この「徹底取材：プログラマの仕事場」という特集なのである．

成功させたのは，ほかならぬビッグブルー王国である．ビッグブルー王国は，1960年代の後半においてシステム360，システム370を完成，汎用コンピュータのアーキテクチャを確立させて強大なコンピュータ帝国を形成した．そこに至って，大量のプログラマが必要とされ，原始プログラマ族が登場したのである．もちろん，プログラミングの世界において結線をかえることによってコンピュータに命令を実行させるという形態が，機械語による記述にかわり，やがてプログラミング言語が誕生したという経過を無視することはできない．プログラミング言語の確立が原始プログラマ族の成立を可能にしたのであった．しかし，原始プログラマ族の使う言語の数は限られたものであった．

図1　原始プログラマの年齢，姿勢，視力の関係
（「コンピュータ考古学会報」，生態考古学の試み，成田・鈴木，1991，Jan.）



帝国の原始プログラマ族

コンピュータ帝国は，中央集権的な階級社会である．これは，コンピュータが大きく高価であったことと無関係ではない．数少ないコンピュータを分割と分業化によって，いかに効率的に使うかが最大の課題であった．昼は，コンピュータ

練習問題②　次の中からまちがった組み合わせをあげなさい．　①DYNAMO－シミュレーション用言語　②PLANNER－人工知能言語　③APL－第４世代言語

利用者の使う時間であって，給与計算や電気料金の計算などに使われた．その計算を行うためのプログラムづくりはもっぱら夜の仕事となる．プログラマ族が夜行性を特徴とするようになったのはその時代に起因している．

分業化が行われており，その業種はまた階級でもあった．ことにプログラマ族にとって，近縁に当たる古代エスイー族との関係は複雑である．多くのプログラマにとって仕事はプログラムの部分的なコーディングでしかなく，プロジェクトの全体像を見るにはエスイー族に昇進するしかなかった．しかし，上級プログラマの中には深夜にコンピュータを占有してプロジェクトを進めるものもいた．そこにパソコンプログラマの源像を見ることができるかも知れない．

マイクロプロセッサ革命と新しいプログラマ社会

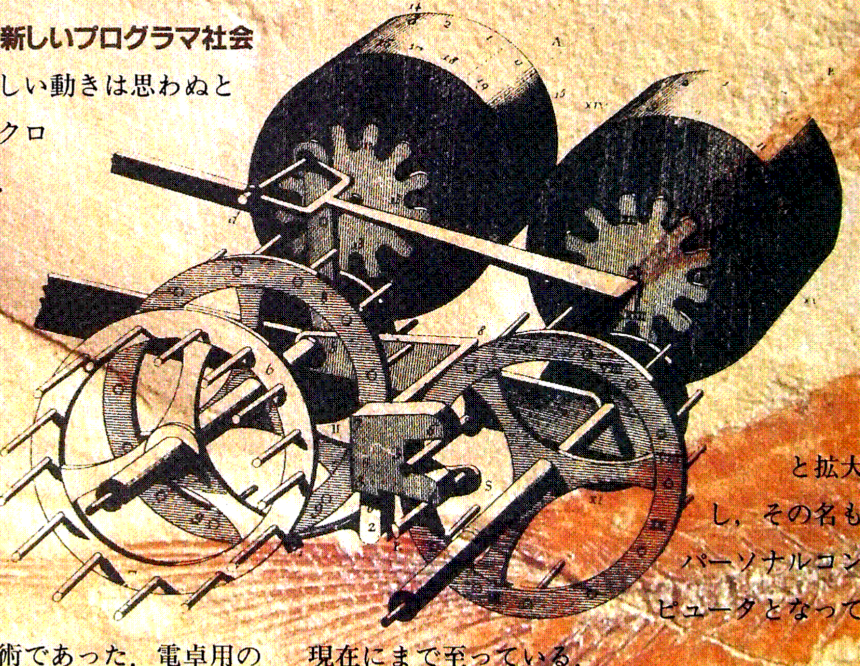
コンピュータ世界の新しい動きは思わぬところから発生した．マイクロプロセッサの出現である．マイクロプロセッサは，重さ数グラムの小さなチップに電子回路を集積する技術によって可能になる．この技術は，それまでの汎用コンピュータからではなく，算盤がわりに簡単に使える計算機，つまり電卓のために開発された技術であった．電卓用のマイクロプロセッサは，ごく小さなコンピュータを作り，それにプログラムをいれることで電卓にするという考え方で実現された．それはそのままごく小さなコンピュータ，マイクロコンピュータの出現を可能にしたのであった．

この出来事は，歴史的にマイクロプロセッサによる革命ととらえることができる．コンピュータは大転換期を迎え，プログラマの可能性は大きく広がることになる．コンピュータの歴史は，先マイクロ期とのあいだに明確な一線がひかれたのである．

プログラマの種族への分化

マイクロプロセッサは，ラジオや無線を扱うことに長けていたマニア族によってマイクロコンピュータに利用された．これは，各地のマニア族から熱狂的な支持を受け，1970年代後半のマイコンブームを花咲かせた．マイコン系コンピュータは，やがてマニア族からビジネス族へと拡大し，その名もパーソナルコンピュータとなって現在にまで至っている．

図2　パスカルの計算機
(『フランス百科全書絵引』，平凡社，1985)

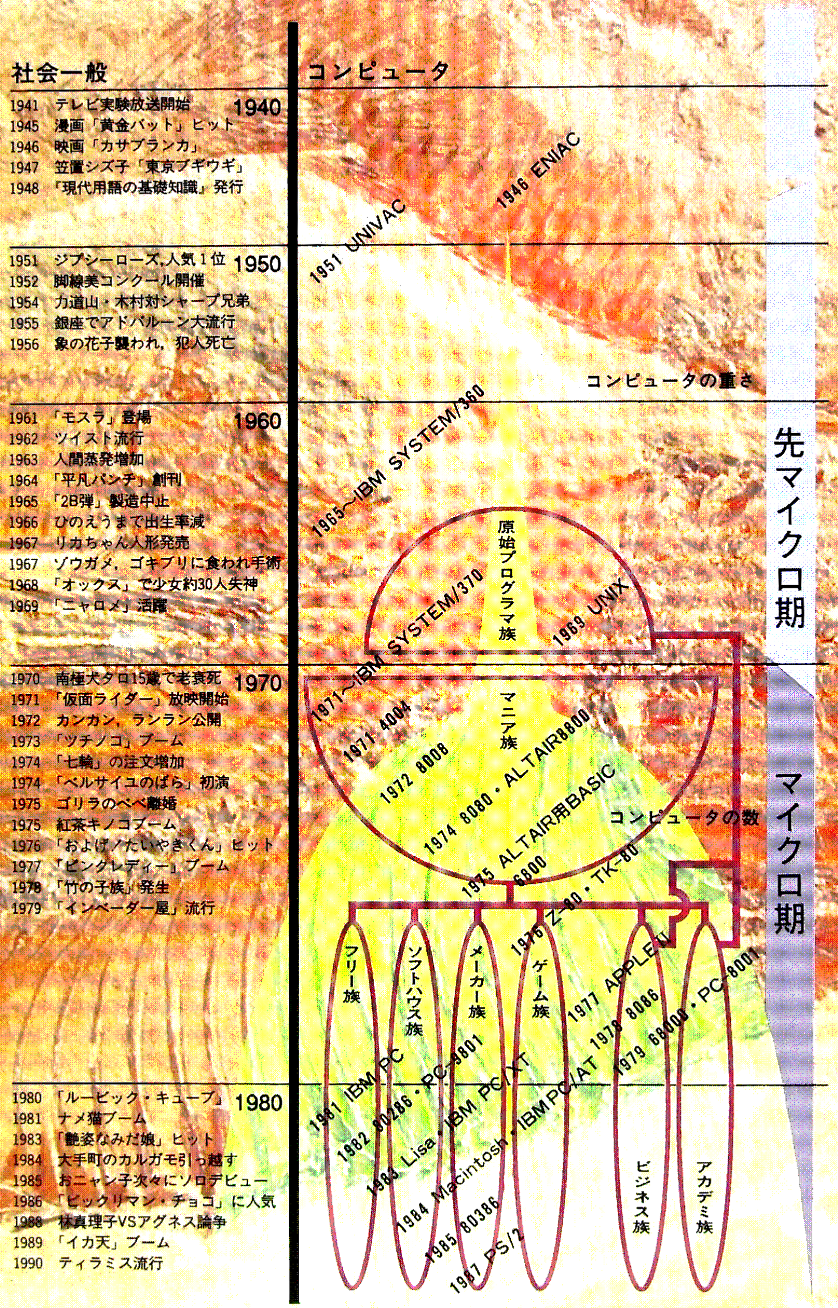


マイコンブーム以降，マイクロプロセッサはコンピュータとしてだけでなく，家庭電機製品の制御にも使われるようになる．いわゆる，「マイコン炊飯器」や「マイコン洗濯機」の出現である．

練習問題③　マイコンと搭載CPUの関係で正しくないものをあげなさい．　①アップルII－6802　②TK-80－μPD 8080 AD　③MZ-80－C－Z 80

マイクロプロセッサは，基本的に「汎用」であるため，プロセッサに何をさせるかはプログラマの仕事である．コンピュータに内蔵するプログラム，プログラムを書くためのコンパイラやインタプリタ，さまざまな機器制御，ワープロや表計算のようなアプリケーションソフトなど，プログラマがやらなければならない仕事は，マイクロプロセッサとパーソナルコンピュータの普及とともに大きく広がった．そのなかで成熟と分化をとげたプログラマ族は，使う道具，言語，行動形態などによってさまざまな分類を行うことができる．

図3　コンピュータの歴史とプログラマ族
(『コンピュータ考古学図説』，コンピュータ考古学会，狭間，1991)



道具による分類を試みた学説によれば，86系と68系という大分類を行ったうえで，PC族，マック族，アミガ族などを論じようとしているが，これは小分類において地域性が強く普遍性を欠いているといわざるを得ない．また，言語による分類は，実質的な共通語であるC言語が広くゆきわたってしまった状況にあっては，大勢をとらえる有効な手段ではない．

コンピュータの歴史を考える上では，ハードウェアやオペレーティングシステム，またメーカーやソフトハウスといった組織の変遷など記しておくべき事柄は多いが，もはや紙幅がつきたようだ．ここでは，生態および棲息地の特徴から分類して，プログラマ社会の各種族を述べることにしよう．

練習問題④　次のうちS-100バスをもつマイコンはどれか．　①TRS-80　②PET-2001　③Cromemco

生態によるプログラマの種族分類

原始プログラマ族● コンピュータ帝国内で，重労働を強いられた初期のプログラマ．おもに計算センターと呼ばれる施設に棲息していた．巨大なコンピュータに仕え，夜間に活動することが多かった．食性は元来雑食であるが，行動の自由が制限されていたため，仕事場を離れることなく食べられるメロンパンやヤキソバパン，カップヌードルなどの食物を好んでいた．こうした環境にあって，その寿命は35年ほどであったと伝えられている．

マニア族● マニア族は，プログラマ族のさまざまな特性を合わせもった強力な能力を有している．多くは，先マイクロ期のラジオ族やオーディオ族の流れを引くものであり，70年代後半のマイコンブームにおいてハンダゴテを片手に活躍した勇者も少なくない．しかし，歴戦の勇者は，マニア族の資質を保ちながらもそれぞれコンピュータの世界に旅立っている．
マニア族の現状はというと，ロリコンゲームマニアから完全なプロフェッショナルではないからとりあえずマニアと呼ばれている人までさまざまである．

フリー族● フリー族の多くはマニア族出身の強者である．ほかの社会におけるフリー族も同様であるが，一般に自由気ままに生きる吟遊詩人のように見れられることが多い．しかし，最近の研究によるとその実態はことなる．優秀な個体ほど忙しく仕事の種類を選ばずに働いたようである．文献や図版によると，優雅な暮らしぶりであるかのように見えることもあるが，それは必死で働くようすを人に見せないように努力している姿か，かっこだけつけて3度のメシも食べられない状況にあったのかもしれないので，疑念をもって見るべきであろう．

ソフトハウス族● この種族は，まことに多様な棲息形態を示す．ソフトハウス族の中でもハケン族は，メーカー族と区別するのが難しい．またフリー族に近い行動形態をとるものもいる．仕事の種類もファームウェアからパッケージソフトまでまちまちだが，所属グループごとにその傾向は決まっている．マニア的傾向をもつものとサラリーマン的傾向をもつものに大別されるが，後者のほうが数の上で勝っているという見解が一般的である．

メーカー族● コンピュータメーカーの研究・開発部門に棲息する．メーカーだから背広姿かというとそうでもない．紺のスーツの課長と頭ボウボウ，Gパン姿の担当者がいっしょに現れたりもすることもあると伝えられている．この場合，Gパン姿のほうはマニア族出身であることが多い．マニア族出身者は，背広を着ていてもヘアスタイルなどの容貌からただのメーカー族でないことは容易に想像がつく．古文献の図版などを観察するときには注意すべきだ．

ゲーム族● 初期のマイクロプロセッサをもっともよく使いこなす種族である．ゲーム族は，マイクロコンピュータの誕生とともに登場し，8080，Z80上のベーシックや機械語を縦横に使う術を身につけしだいに複雑なゲームを作り出すようになった．ほぼすべての分野で活動し，幾多の成功物語を生み出しているが，棲息する環境は改善されていないようだ．この種族には，ニンテン王国の支配権が広くおよんでいる．

ビジネス族● マイクロプロセッサによるPC文化の誕生によって，もっとも恩恵を受けたのはビジネス族であろう．夜しかコンピュータを使えないという状況は改善され，いつでもコーディングを行えるようになった．コンピュータを導入し，夜な夜なプログラムを組んでいるうちに，コンサルタントと呼ばれる予言者になったり，ソフト会社の長となったものも少なくない．行動は，マニア族に似ているが，Gパンを衣装としたマニア族にたいして，ビジネス族はスーツにネクタイという衣装が多い．

アカデミ族● おもに大学や研究所に棲息する．ビジネス族などに比べるとその生態は複雑である．使っているコンピュータの種類は豊富で，先マイクロ期の大型機からワークステーション，98，絵文字を使うマッキントッシュになどさまざまである．大型機とワークステーションで研究活動を行い，98で論文を書き，マッキントッシュでゲームをするという多様な行動パターンをとることも珍しくない．文献を作るのが仕事であるため資料は豊富である．

練習問題の答　①－②1951年にユニバックから発売された　②－③APLは統計や心理分野で使われる　③－①アップルⅡは6502　④－③

徹底レポート

プログラマの仕事場

Programmer's Working Space

「パソコンプログラマのイメージを具体的に描き出してみよう」というのが，一連の取材の目的だ．もちろん現実から提示されるイメージは一様なものではない．仕事場の規模によって，あるいはプロジェクトの段階によって，仕事場はさまざまに形態を変化させる．フリーのプログラマは，パーソナルという面にこだわっていかにして快適な開発環境を築くかがテーマになるし，複数の人間がプロジェクトにかかわる仕事場では共同で開発に当たるためのノウハウが蓄積される．しかし，どの仕事場でもパーソナルなシステムが基本になっていることが感じられる．たった1人の仕事場から100人をこえる開発部門まで，パソコンプログラマが活躍する現場を見てみよう．

プログラマの仕事場 | 1

PW Programmer's Working Space

イースト株式会社

Windows, プレゼンテーションマネージャ関連のソフトウェアを手がけるプログラマーズカンパニー

Windows 3.0が，プロテクトメモリの利用を可能にするなどの大幅な機能アップを伴って登場したのは，1990年5月のことだった．91年には，いよいよ日本語版Windows 3.0が発表されるという．次代のOS環境として期待の高まるWindowsだが，国内では，本格的にWindowsアプリケーションに取り組んでいるソフトウェアメーカーは，まだそう多くはない．イースト株式会社は，3年以上前からWindowsに照準をあわせ，Windowsアプリケーションの開発に取り組んできた．WindowsやPresentation Managerといった，新しい環境でのプログラミングの現場について，同社営業部取締役部長下川和男氏にお話をうかがった．

洗練されたユーザーインターフェースを備えた期待のOS環境として，Windows 3.0が華々しく登場した今となってみれば，3年前からWindowsアプリケーションを手がけていたというイーストには，先見の明があったというべきだろう．だが，Windowsアプリケーションを始めるにあたっては，いろいろとお家の事情があったようだ．

イーストが設立された6年前というのは，ちょうどあの「一太郎」が発売されたのと同じ年にあたる．これを機に，MS-DOS上で動くアプリケーションが続々とパソコン市場に登場することとなる．先行するジャストシステム（「一太郎」）や管理工学研究所（「松」）ほどMS-DOSでのソフトウェアの蓄積がないということは，後発のソフトウェアメーカーにとって大きなハンディである．それならば，MS-DOS上のソフトウェア資産が役に立たないWindowsで勝負をしようというのが，イーストの選択した道だった．

先行するソフトウェアメーカーに挑むためには Windowsという新しい環境に取り組む必要があった

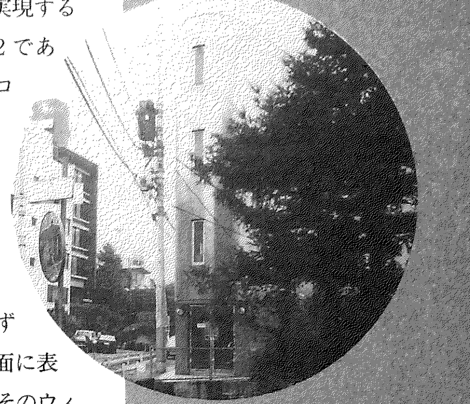
Windowsプログラミングとはいかなるものか？　ここでひとつ簡単に紹介してみよう．Cを使って画面に"hello world"と表示するプログラムは，読者のみなさんもよくご存じのことと思う（リスト1）．

リスト1　DOS版 hello world

```
main(
    printf("hello world¥n");
}
```

同じ内容をWindows上で実現するためのプログラムがリスト2である．どうしてこれほど大量のコードになってしまうのか？実は，このソースコードのほとんどの部分はWindowsシステムの設定にかかわるものだ．ウィンドウクラス，ハンドルなどをセットして，まずWindowsにウィンドウを画面に表示するよう指示する．次に，そのウィンドウの中に文字を表示するわけだが，Windowsプログラミングでは，メモリ，ディスプレイ，キーボード，マウス（これらをまとめてWindowsでは「リソース」と呼ぶ）などの管理は，すべてWindowsのシステムを通して行うよう決められている．したがって，たとえ「文字列を画面に表示する」という簡単な内容の処理でも，見慣れたprintf関数ではなく，Windows独自の関数を使うことになる．

このことからわかるように，演算処理などの少数の例外を除いて，Windowsアプリケーションの開発にはそれまでのMS-DOS上のプログラミングで蓄積してきたソフトウェア資産は利用できない．同じCやアセンブラを



Location

本社は渋谷区代々木1-3-1．JR代々木駅前から徒歩5分程，明治神宮の北鳥居に近い一角にある4階建のビルである．代々木とはいっても，予備校生でにぎわう駅前とは違って，辺りはこじんまりしたオフィスが多い．このほかに，千駄ヶ谷，南新宿の2か所にオフィスを構える．

About Company

イースト株式会社，丸山雄三社長．

昭和60年社員8名で設立．OEMを中心としたソフトウェア開発で急成長してきた．現在は，社員94名で，そのうち86名がプログラマである．社長と経理，総務を除いたほとんどがプログラマか技術者．取締役でも，机の上にマシンを3台ずつ並べてコーディングするという「プログラマーズカンパニー」である．

最近ではOS/2のPresentation Manager（PM）を使って，パソコンをメインフレームとリンクして業務処理を行う大規模なシステムのうち，パソコン部分のソフトウェア開発だけを依頼されることも多く，Windows，PMのスペシャリストとしての地位を着々と築きつつある．

Products

MS-Windows対応日本語ワープロソフト
TextWriter
OS/2 PM対応日本語ワープロソフト
TextWriter/2

使うとはいえ，通常のMS-DOS上のプログラムとはまったくの別物というわけだ．

✚

他社と同じ立場で競争しようと取り組んだWindowsアプリケーションだが，当初の苦労は並大抵のものではなかった．最初に開発したWindowsアプリケーションは，納品するときにもマウスによる操作のみが可能で，まだキーボードからの入力ができなかったという．下川氏は次のように当時を振り返る．

どっちを向いてもわからないことだらけ!? プログラマ泣かせのWindows

「最初の1年間というのはわからないことだらけで，とりあえず作ってみるということの繰り返しでした．SDK（注1）の中にマニュアルは入っていたんですが，非常に読みづらいし，関数のリファレンスマニュアルも，難しい関数ほど何を書いているのかわからなくて困ったのを覚えています．わからないところは自分で一連のサンプルプログラムを作ってテストしてみる以外に手がないんですね」．

「次の1年はそれがだんだんわかってくる時期で，米国で『PROGRAMMING WINDOWS』（注2）という本が出まして，これを10冊程買いました．プログラマはずっとこれを読んでましたね．次の年には，『ウィンドウズ・プログラミング』（注3）なども出たのですが，その時点ではその本しかありませんでした．

その1年である程度Windowsにも慣れてきました．やがて，Windowsや

リスト2　Windows版 hello world

```
#include <windows.h>          /* Windows用ヘッダファイル */

long FAR PASCAL HWndproc(HWND, unsigned, WORD, LONG);

int PASCAL WinMain(hInstance, hPrevInstance, lpszCmdLine, CmdShow)
HANDLE hInstance;             /* インスタンス・ハンドル */
HANDLE hPrevInstance;         /* 直前のインスタンス・ハンドル */
LPSTR lpszCmdLine;            /* コマンド行の引数へのポインタ */
int CmdShow;                  /* ウィンドウの表示形式 */
{
    HWND hWnd;                /* ウィンドウハンドル */
    MSG  msg;                 /* メッセージ */
    WNDCLASS wndClass;        /* ウィンドウ・クラス */

    if (!hPrevInstance) {     /* 最初のインスタンスなら */
        wndClass.hInstance     = hInstance;
        wndClass.lpfnWndProc   = HWndproc;
        wndClass.hIcon         = LoadIcon(NULL, IDI_APPLICATION);
        wndClass.hCursor       = LoadCursor(NULL, IDC_ARROW);
        wndClass.hbrBackground = GetStockObject(WHITE_BRUSH);
        wndClass.lpszMenuName  = (LPSTR) NULL;
        wndClass.lpszClassName = (LPSTR) "HELLO";
        wndClass.style         = CS_HREDRAW | CS_VREDRAW;
        wndClass.cbClsExtra    = 0;
        wndClass.cbWndExtra    = 0;

        if (!RegisterClass(&wndClass)) {
            return (NULL);   /* クラス登録に失敗したら終了 */
        }
    }

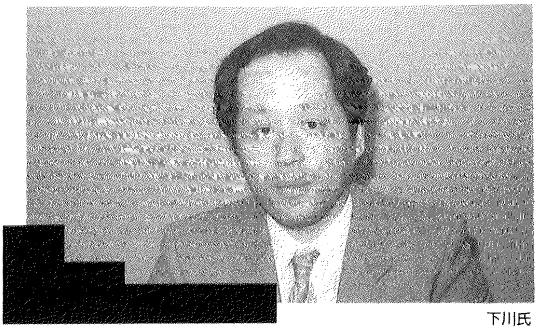
    hWnd = CreateWindow(            /* ウィンドウを作成してハンドル取得 */
            "HELLO",                /* ウィンドウ・クラス名 */
            "hello",                /* ウィンドウ名 */
            WS_OVERLAPPEDWINDOW,    /* ウィンドウスタイル */
            CW_USEDEFAULT,          /* 初期 x 位置はデフォルト */
            CW_USEDEFAULT,          /* 初期 y 位置はデフォルト */
            CW_USEDEFAULT,          /* ウィンドウの幅はデフォルト */
            CW_USEDEFAULT,          /* ウィンドウの高さはデフォルト */
            NULL,                   /* 親ウィンドウのハンドル */
            NULL,                   /* メニューのID */
            hInstance,              /* インスタンス */
            NULL);                  /* 付属情報 */

    ShowWindow(hWnd, CmdShow);      /* メイン・ウィンドウを表示 */
    UpdateWindow(hWnd);             /* WM_PAINTメッセージを送る */

    while (GetMessage(&msg, NULL, 0, 0)) { /* メッセージループ */
        TranslateMessage(&msg);     /* キャラクタメッセージを変換 */
        DispatchMessage(&msg);      /* メッセージを分配 */
    }
    return (msg.wParam);
}

long FAR PASCAL HWndproc( hWnd, Message, wParam, lParam)
HWND hWnd;                     /* ウィンドウ・ハンドル */
unsigned Message;              /* メッセージ識別番号 */
WORD wParam;                   /* 付属情報 */
LONG lParam;                   /* 付属情報 */
{
    HDC hDC;                   /* デバイス・コンテキスト・ハンドル */
    PAINTSTRUCT ps;            /* 描画情報 */

    switch (Message) {
    case WM_DESTROY:     /* ウィンドウが消されたとき */
        PostQuitMessage(0);
        break;
    case WM_PAINT:       /* ウィンドウ再描画 */
        hDC = BeginPaint(hWnd, &ps);
        TextOut(hDC, 1, 1, "hello, world", 12);
        EndPaint(hWnd, &ps);
        break;
    default:             /* Windowsのデフォルト処理 */
        return (DefWindowProc(hWnd, Message, wParam, lParam));
    }
    return (NULL);
}
```


下川氏

PM (Presentation Maneger)のプロジェクトが10とか20とか社内で走り出すと，エンジニアのチーフ同士がお互いにわからないところを質問しあって，話ができるようになりました．たとえば，プリンタドライバ部分についてはアイツに聞けばわかるとか，ベクタグラフィックスなら誰それということが社内的にわかってくると，お互いに相談できるわけですね．それでずいぶん楽になりました．

任された技術者が，『もう辞めたい』といってきたこともあります．『こんなのどうやっていいかさっぱりわからない』という話もありました」．

(注1) SDK
Windows Software Development Kit，Windowsソフトウェア開発キットのこと．メーカーによってはWindowsツールキットという呼び方もある．

(注2)「PROGRAMMING WINDOWS」(CHARLES PETZOLD著，米国Microsoft Press発行)
アスキー出版局発行の『プログラミングWINDOWS』(Volume 1：4120円，Volume 2：3610円) はこの邦訳．原書は1冊だが，訳書はVolume1，Volume2の2分冊でそれぞれ約600ページほどある．Microsoft Windows 2.0とWindows/386の下で動作するプログラムをC言語で書くための解説書．

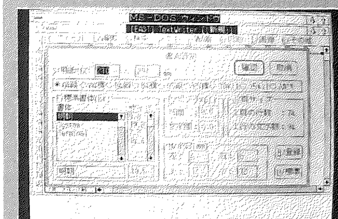
(注3)『ウインドウズ・プログラミング』(David Durant，Geta Carison，Paul Yao著，富士ソフトウエア教育出版部訳・編，富士ソフトウエア，3914円)
米国SYBEX社発行の『Programmer's GUIDE To WINDOWS 2nd Edition』の邦訳．Windowsプログラミングに必要な概念の解説にはじまり，プログラミングに必要な知識を網羅してある．

3～20人のチーム単位で開発プロジェクトを担当
チームリーダーとなる主任クラス役割が大きい

PMとWindowsアプリケーションの開発がイーストの技術的な面の特徴だとすると，会社組織としての特徴は「技術のトップ」にあたる人間が特に存在しないことだ．ジャストシステムの浮川初子氏やサムシンググッドの長谷川氏のような「技術のトップ」が会社の創業時からソフトウェア開発の先頭に立ち，いわば徒弟制度のようにしてプログラミング技術を教えるという形ではなく，最初から複数のトップ技術者がいたのだという．

その影響なのか，「プログラムはこういう風に作るべし」といった特別な規約が社内にはない．プロジェクトごとのルール作り，共通のライブラリの管理，どんな風にコーディングするかなどは，すべてそのプロジェクトのリーダー（主任）の仕事になっている．主任は現在社内に約20名．それぞれが1

OS/2 PM対応表計算ソフト
　WinCalc
日本語ワードプロセッサ　EW
かな漢字変換フロントプロセッサ　E1
AVマルチメディアエディタ
　HyperUD
ISAMファイルアクセスライブラリ
　EISAM
各種C言語ソースプログラムライブラリ(かな漢字変換ライブラリ，汎用スクリーンエディタ，PostScriptライブラリ，TIFFイメージ編集ライブラリ，グラフ表示サブシステムなど)
そのほかにも各種プログラム開発支援ツールがある．


TextWriter/2

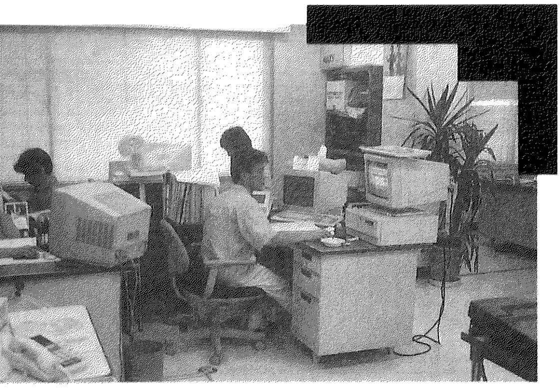
MS-Windows

MS-DOSにマルチウィンドウ表示，マルチタスクといった新しい操作環境を付加するソフトウェア．画面の表示にはカラーグラフィックスを利用し，ポップアップメニュー，アイコンなどの視覚的なイメージを使って，利用者にわかりやすい操作方法を提供する．MS-Windowsのような操作環境はGUI（グラフィカルユーザーインタフェース）と呼ばれ，次世代のパソコンには欠かせないといわれる．

アプリケーション開発の点から見ると，デバイス(ディスプレイ，キーボード，マウス，プリンタ，I/Oポートなど)に依存する部分の管理はすべてWindowsシステムにまかせることができるため，機種に依存しないプログラムを作成することができる．

米国では1990年5月にMS-Windows 3.0が発表されて，従来8086/80286マシン用Windows/286と80386マシン用のWindows/386の2つに分けられていた製品が一本化された．日本語対応バージョンではMS-Windows 2.1が最新．

つのプロジェクトを任されていて，その下に数人のプログラマがつくチーム制をとっている．各チームの人数は，プロジェクトの内容に応じて，3，4人の小規模なものから20人の大所帯までさまざまだ．



では，具体的にどんな形でプロジェクトを進めているのだろうか．

「まず，営業活動をして仕事を取ってきます．普通はお客様からこんなことをやりたいという内容を書いたものをいただきます．MS-DOS版の仕様書があって『これのWindows版を作ってほしい』という話も多いです．その場合は社内で画面デザインをして，Windows版ではこうなりますというのをの外部仕様書にして，コーディング，テストとなります．基本設計からまかされるというプロジェクトも増えています」．

「お客さんの方で開発に失敗して，その結果ウチに回ってくるプロジェクトというのもあります．『Windowsのコ

COLUMN

Windowsアプリケーションの仕組み

1．基本的なWindowsアプリケーションの開発環境

Windows上で動作するアプリケーションを作成するためには，まず，Windowsプログラミングの開発環境を整えなくてはならない（表A参照）．

ハードウェアとしては，Windowsを動かすためのパーソナルコンピュータ，カラーモニタが必要となる．本体メモリは640KB以上，CodeViewデバッガを使用するには，さらにLIM4.0対応のEMS拡張メモリが必要だ．ハードディスクは，できれば40MB程度の容量があることが好ましい．ツールキットに含まれる開発ツールの中には，マウスがないと動作しないものがある．

Windowsシステムの実行中には，Windowsが画面の制御を占有してしまうので，デバッグ用にもう1台パーソナルコンピュータが必要となる．シリアルポート経由で2台のコンピュータを接続し，SYMDEBやCODEVIEWデバッガの入出力をデバッグ用ターミナルにリダイレクトする．

ソフトウェアとしては，Windowsツールキット（SDK）が必要．これは，Windows専用の開発ツール集で，Windows用ヘッダファイルやライブラリ，各種開発ツールが含まれる．表B中，Cコンパイラとマクロアセンブラを除くすべてのツールはこのWindowsツールキットに含まれている．

Cコンパイラは Windows対応のものであれば何でもよいが，日本語環境では実質的にMS-C 5.1以上ということになる．またCでアプリケーションを作成するのであれば，マクロアセンブラはなくても構わない（ただしWindowsライブラリを開発する場合は必須）．

ハードウェアの準備が整ったら，Cコンパイラ，Win

表A：Windowsアプリケーション開発に必要なもの

ハードウェア：
- Windowsが動作可能なパーソナルコンピュータ
- 1台以上のフロッピーディスクドライブ
- ハードディスク
- マウス
- EMS拡張メモリ
- デバッグ用ターミナル
- RS-232Cケーブル

ソフトウェア：
- Windowsツールキット（SDK:Software Development Kit）
- MS-DOS（3.0以上）
- Cコンパイラ
- MS-Windows
- マクロアセンブラ

表B：アプリケーションの開発に使用するツール

Cコンパイラ	CL
リソースコンパイラ	RC
リンカ	LINK4
アイコンエディタ	ICONEDIT
ダイアログエディタ	DIALOG
プログラムメンテナンスユーティリティ	MAKE
シンボリックデバッガ	SYMDEB
コードビュー	CVW
マクロアセンブラ*	MASM
フォントエディタ**	FONTEDIT

*　Windowsのライブラリを開発する場合に使用
**　フォントリソースを開発する場合に使用

PM, Windowsソフトへクライアントの要望は多いが実態の認知度はいまひとつ?

ーディングがこんなにめんどくさいと思わなかった』というわけです．こういう場合は，４カ月先とか納期が近づいているのが多くて結構大変ですね．

普通は４，５人規模の仕事なら，半年で１プロジェクトが終ります．お客さんの方で仕様まで決まっていると，ときどき非常に困ることもあります．前にPM上のプロジェクトでダイアログボックスでの項目ごとの移動をリタ

OS/2 PM (Presentation Manager)

OS/2のグラフィカルユーザーインターフェース．画面の表示や操作方法，プログラミングの考え方はMS-Windowsとよく似ているが，MS-WindowsがMS-DOS上で動くソフトウェアであるのに対し，PMはOS/2そのものの一部として組み込まれている．PMが提供されるのはOS/2 1.1以降のバージョン．

---

dows，Windowsツールキットをインストールする．ひとつ注意しなければならないのは，Windowsはデバッグバージョンのものを使用するということ．デバッグバージョンにするためには，Windowsのオリジナルのフロッピーディスクのバックアップを取り，これにWindowsツールキットから３つのファイルをコピーする．この新しいフロッピーディスクを使ってインストールすればよい．

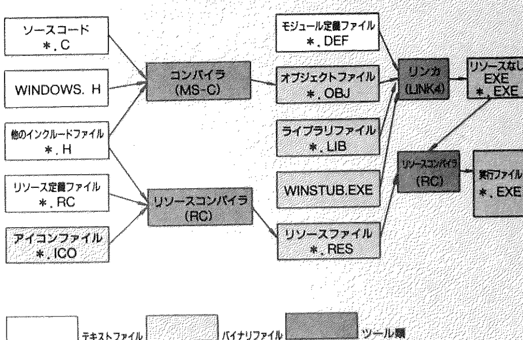
最後にAUTOEXEC.BATファイルを変更して，これらのディレクトリのパス指定を行えば，準備完了だ．

2．Windowsアプリケーションの開発手順

Windowsアプリケーションと通常のDOSアプリケーションの開発手順が大きく異なる点は，リソースの取り扱いである（図A参照）．Windowsでは，アイコン，メニュー，カーソル，ダイアログボックス，フォントなどのアプリケーションプログラムが使用するさまざまなオブジェクトをリソースと呼び，プログラムのソースファイルとは別に，リソース記述ファイルに定義する．そして，リソース記述ファイル専用のコンパイラ（リソースコンパイラ）を使ってコンパイルし，別に作成した実行可能ファイルにつけ加える．

リソースだけを別ファイルとしてコンパイルするのは，Windowsがマルチタスクシステムであることと深い関わりがある．そもそも640KBという限られたメモリの中でMS-DOSとWindowsシステムが占める領域を除くと，せいぜい200KBから300KBしか残らない．そこで複数のアプリケーションが動作するとなると，Windowsは常にメモリ配置を監視して，これを最適にしなければならない．リソースファイルとして記述されたオブジェクトは使用される時に初めてロードされるのだが，このときプログラムのデータセグメントとは別にリソース専用のセグメントが用意される．このことによってWindowsシステムは，メモリ配置を管理しやすくなり，また同じプログラムが複数動作する場合には，メモリを共用することができるというメリットもある．

図A：WindowsアプリケーションとDOSプログラムの作り方の違い



①テキストエディタを使って，Cまたはアセンブラでソースファイルを作成する．
②Cのソースファイルは Cコンパイラを使ってコンパイル，アセンブラのソースファイルはマクロアセンブラを使ってアセンブルする．
③アイコンエディタを使ってカーソル，アイコン，ビットマップを作成する．
④ダイアログエディタを使ってダイアログボックスを作成する．
⑤テキストエディタを使って，リソースを定義したリソース記述ファイルを作成する．
⑥リソースコンパイラを使ってリソース記述ファイルをコンパイルする．
⑦テキストエディタを使って，モジュール定義ファイルを作成する．
⑧コンパイルされたソースファイル(オブジェクトファイル)，ライブラリファイル，モジュール定義ファイルをリンクし，実行可能形式のファイルを作成する．
⑨リソースコンパイラを使ってコンパイルされたリソース記述ファイルを実行可能形式のファイルに加える．

ーンキーでやってほしいというのがありました．PMのダイアログエディタを使うのであれば，項目間の移動はTABと決められているんです．TABを使えばダイアログエディタで簡単に作れたものを，しょうがないんではじめから全部コーディングしました．PMとかWindowsの作法（注4）というのがあるんで，それを無視した要求っていうのは困りますね．でもお客さんのいうことは聞かざるをえないんですけれど」（下川氏）．

PMやWindowsのプログラムでは，画面に表示されるウィンドウやアイコン，メニューなどはリソースエディタ(アイコンエディタ，ダイアログエディタ，フォントエディタ)を使って処理の手続きとは別に書くことができる．この機能を活かしてプロトタイプを作り，クライアントに見せて了承を得ることもある．この方法によれば，ウィンドウやメニューが表示され，外見上はWindowsアプリケーションができあがってしまったかのように見える．アプリケーションの仕様を説明するためには格好のものではあるが，これには思わぬ副産物もあるようだ．↙

開発用とターゲットマシンなどひとりに2,3台なければ仕事にならない

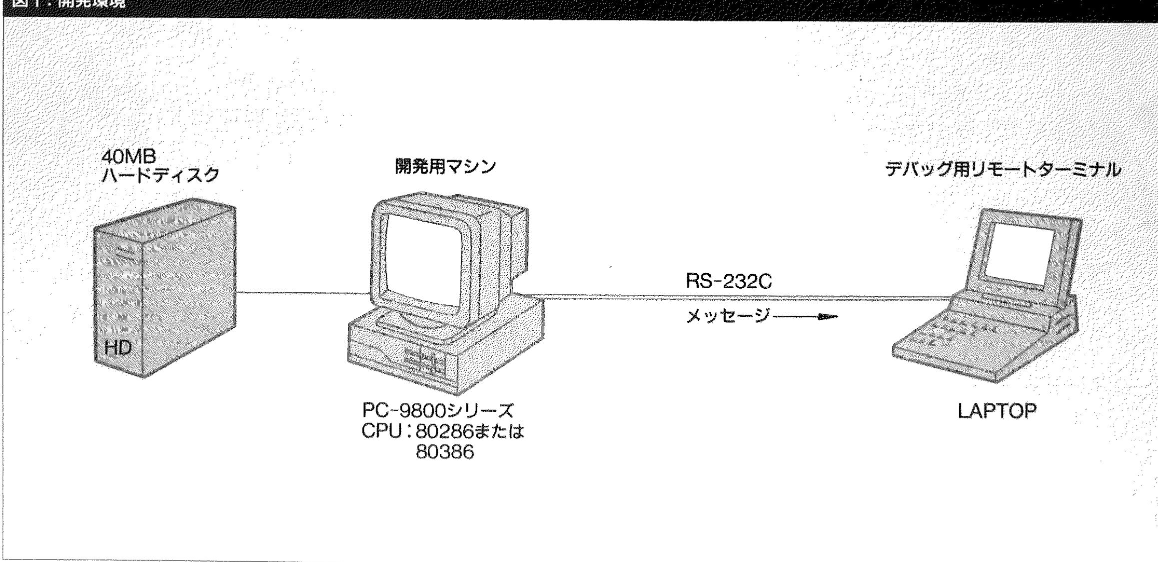
「具体的になりわかりやすくなった分，お客さんも苦情がいいやすくなるんです．ここはもっとこうして欲しいとか．かえって仕様変更の要望が増えてしまいました」．

では，代表的な開発環境を紹介しよう（図1参照）．PM，Windowsのアプリケーション開発が中心のイーストでは，基本的にひとりのプログラマにつき，開発用とデバッグ用のパーソナルコンピュータがそれぞれ1台ずつある．開発用のマシンは，CPUが80286または80386で40MBのハードディスク付き，機種はおもにNECのPC-9800シリーズである．デバッグ用にはPC-98LTなどを使っている．

PM，Windowsともにアプリケーション自体は機種に依存しないので，中には実際に開発の対象とする機種がNECのPC-9800シリーズ以外の場合でも，使い慣れた環境にこだわるプログラマもいるという．このような場合には，プログラミング用のマシンのほかに，実行用，デバッグ用と合計で3台のマシンを占有することになる．最近ではOS/2関連の仕事

（注4）PM，Windowsの作法
「アプリケーションスタイル」と呼ばれる．ウィンドウを構成する要素，操作するキーなどが決められている．こうした約束ごとは面倒に感じるかもしれないが，アプリケーション操作の共通性を追求するには不可欠なものだろう．

図1：開発環境



が増えてきたため，「CPUは80386の20MHz，ハードディスクは80MB，ディスプレイはハイレゾリューションを」と望む声も徐々に大きくなっているとか．ひと昔前までは紙と鉛筆さえあればすぐにできると陰口をいわれたソフトウェア会社だが，90名近いプログラマ全員にこの環境を用意するとなると，設備にかかる費用も馬鹿にはならないようだ．

ソフトウェアの方ではCコンパイラがMS-C 5.1，これにPMまたはWindowsのSDK．ディレクトリの構成をたずねると「ごく普通にインストールしているだけ」という（図2参照）．エディタは，プログラマがそれぞれ自分の好みのものを使用している．人気があるのはMIFES，VZ，それにREDである．メモリの利用方法についても，まったくのプログラマまかせである．

図2：ディレクトリ構成（Windowsの場合）

```
─┬─BIN          Windowsの基本ツールを含むディレクトリ
  ├─BINB         OS/2，MS-DOS共用の実行ファイルを含むディレクトリ
  ├─BINR         MS-DOSでのみ実行可能なファイルを含むディレクトリ
  ├─INCLUDE      インクルードファイルを含むディレクトリ
  │  └─SYS       システムインクルードファイルを含むディレクトリ
  ├─LIB          ライブラリファイルを含むディレクトリ
  ├─SOURCE
  │  ├─DOC       ドキュメントファイルを含むディレクトリ
  │  └─ME
  │     └─INI    エディタの初期化ファイルを含むディレクトリ
  ├─TMP          テンポラリファイルを作るためのディレクトリ
  └─WINDOWS      Windowsシステムを含むディレクトリ
     └─PIF       PIFファイルを含むディレクトリ
```

Actorなどの特別な開発ツールは使っていない．社内では特別なソフトウェア資産管理や，ライブラリの共通化もしていない．エディタと基本ツールのほかには，小さなツール類をいくつか作った程度だという．ちょっと意外な感じがするが，「ソフトウェアの開発というのは，結局個人にしかノウハウが残らない部分がありますね．それに，社内のプログラマ，とくに主任レベルの力量はわかっていますから」と，あくまで会社の財産は「プログラム」ではなく「プログラマ」であるという方針だ．

その一方で，「社内で共通に使えるライブラリがあれば，これはもう商品として販売してしまいます．そこまでのレベルのライブラリはなかなかないですけれど」（下川氏）と，常にソフトウェアの製品化をねらっている．

✚

「特別な開発ツールもない，コーディングの規約もない」というイーストだが，下川氏が「ひとつだけぜひ紹介↙

自社開発のツールDocGenでクライアントに提出するドキュメントを作成

しておきたいツールがある」といって見せてくれたのが，「プログラム仕様書生成ツールDocGen」だ．イーストはパッケージソフトの開発もするが，クライアントから直接依頼されて，ソフトウェアを作ることも多い．このようなオーダーメイドのソフトでは，納品時に仕様書の提出を求められるのが通例だ．

ところが，仕様にはいつも変更がつきものである．モジュール構造や共通変数の一覧，関数の仕様書などは，プログラムを組む前に作ることにはなっているが，必ずしも最初の仕様通りのプログラムにはならないこともしばしばある．そこで，プログラムのコメントの付け方に規約を設けて，あるコメントが入っていたらそれをもとにして関数一覧表，関数仕様書，モジュールのCALL，CALLEDの関係，関数や外部変数のクロスリファレンスを出力できるようにした．現在は8種類のドキュメント生成ツールがある（表1参照）．

仕様が変わるたびにプログラムと仕

表1：プログラム生成ツールDocGenの機能

プログラム名	処理の概要
CTOFUNC	Cのソースプログラムから関数一覧表を生成する
CTODOC	Cのソースプログラムから関数仕様書を生成する
CTREEX	Cのソースプログラムからモジュール関連図を生成する
XREF	Cのソースプログラムから関数名クロスリファレンスを生成する
VXREF	Cのソースプログラムから共通変数名クロスリファレンスを生成する
STEP	Cのソースプログラムからステップ数一覧表を生成する
DOCTOC	関数仕様書からソースプログラムのコメントを生成する
P, PP	ソースプログラムリストを印刷する

・MS-DOSまたはOS/2が稼働するパーソナルコンピュータであれば，メーカー，機種を問わず利用可能

図3：プログラマの作業手順とDocGenの利用

プログラマの作業	利用するツール名
① 関数仕様書を作成する	
↓	
② ①からソースプログラムのコメント部分を生成する	DOCTOC
↓	
③ プログラムをコーディングする	
↓	
④-(1) 完成したプログラムから関数仕様書を作り直す	CTODOC
-(2) 関数一覧表を生成する	CTOFUNC
-(3) モジュール関連図を生成する	CTREEX
-(4) 関数クロスリファレンスを生成する	XREF
-(5) 外部変数クロスリファレンスを生成する	VXREF
-(6) ステップ数を計算する	STEP
↓	
⑤ ソースプログラムを印刷する	P, PP

様書の両方を変更するのは，プログラマにとってかなりめんどうな仕事だが，このDocGenを使えば，仕様書とソースプログラムの一元管理が可能となる．ソースプログラムを変更したついでに，コメントを修正しておけば，あとはツールを通すだけで新しいドキュメントができるという仕組みだ（出力例参照）．実際にはプログラマの作業手順にそって，個々のツールを使うよう推奨されている（図3参照）．

8つのツールは，それぞれにソフトウェアメーカーならではの細かい工夫が見られる．たとえばプログラムのステップ数一覧を出力するプログラムSTEPを見てみよう．通常ソフトウェアの見積りは，「このシステムは全部で約3万ステップだから，1ステップ○円で合計××円」という具合いにステップ数単位で計算する．ところが，このステップ数の数え方が難しい．コメント行は除くのかどうか，関数の引数を1行にひとつずつしか書かない場合も1ステップと数えるのか（PMではこの書き方が標準である）どうかなど....プログラムのステップ数をどう数えるかは，そのまま売上に関わるだけに大問題である．そこで，STEPはただ単にソースプログラムの行数をカウントするだけでなく，コメント行数や，コメント率を表示したり，オプションで()内の，(カンマ)を1ステップとしてカウントするかどうかを指定することもできるようになっている（出力例1参照）．

下川氏は，「MS-C 6.0には，DocGenのツールと同様の機能が含まれるという噂もあるので，それとの関係を見極めてから」という条件付きで，DocGenをパッケージソフトとして発売することも検討しているとも語ってくれた．

さて，日本でも期待の高いWindows 3.0だが，イーストではこの新しいWindowsにはどのように取り組んでいくのだろうか？

「現在の仕事の割合はPM 7割，Windows 2割，その他1割というところですが，最近ではWindowsが増えています．アメリカで3.0が発表になって，日本でもやってみようという企業が増えつつありますね．TextWriterについては，作成したときと同じ体制でプロジェクトは続けていま

出力例1：DocGen出力例

関数名	ファイル名	位置	行数	コメント数	ステップ数	コメント率
合計	ejeext.h	0:	65	8	47	14%
_get_wval	ejecom.c	59:	65	11	44	20%
合計	ejecom.c	0:	65	11	44	20%
main	ejdemo.c	53:	80	7	67	9%
test2	ejdemo.c	87:	16	3	8	27%
test1	ejdemo.c	109:	49	0	60	0%
jikoho	ejdemo.c	158:	7	0	4	0%
総合計		0:	5992	1250	3556	26%

すので，まずは性能アップして，次に3.0対応にしようと考えています」。

バージョンアップの際に必ず問題になるのが，前のバージョンとの互換性だが，新しいWindowsではどうなのだろう？

Windows 3.0で異なるのはメモリ管理だけ プログラマが楽をできる環境が継承された

「Windows 2.1と3.0はメモリ管理のやり方が違うだけなんです．昔はWindowsとPMでソースが変わらないという話があって，それは嘘でしたが（笑）今度は大丈夫です．試してみましたが，プリンタドライバのいくつかはそのまま動きました．カナ漢字変換などはリコンパイルしたら動きました．3.0の仕様をしっかり理解できれば，大きなプログラムでも2.1でも3.0でも動く形で作れるでしょう．「MS-Word for Windows」（注5）なんかはそのまま動きます．

API（注6）で変わったものがほとんどありませんから，ソースとしては何も変わらないんです．プログラマに楽をさせてやろうという考え方はずっと生きています．やっぱりビル・ゲイツ（注7）は信じておいていいなぁと思いましたね」。

（注5）MS-Word for Windows
Windows上で動作するワープロソフトの名称．Windowsの開発元である米国Microsoft社の製品．

（注6）API（Application Program Interface）
一般に，アプリケーションがOSの機能を利用するために呼出すコマンド群のことを指す．C言語でWindowsアプリケーションを開発するときには，API関数と呼ばれる関数を使用する．

（注7）ビル・ゲイツ（Bill Gates）
現米国Microsoft社会長．同社の設立者のひとりでもある．

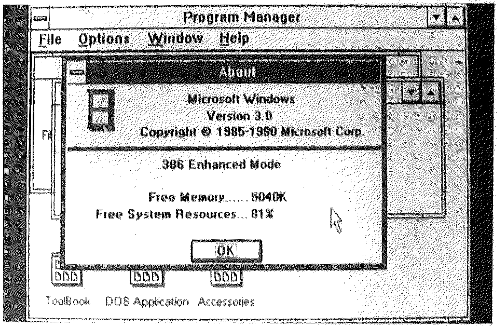
COLUMN

Windows 3.0

Microsoft Windows 3.0は，1990年5月，アメリカをはじめ世界各地で一斉にリリースされた（残念ながらこの「世界各地」に日本は含まれていない）．OS/2 1.2のPresentation Managerライクなグラフィカルインターフェース，メモリ管理機構の改善，ネットワーク対応など，これまでのWindowsの欠点を補ってあまりある内容との評判が高い．

従来8086，80286マシン用のWindowsと80386用のWindows/386の2つに分かれていたWindowsは，このWindows 3.0に一本化された．8086の機能に依存し，旧Windowsアプリケーションとの互換性を保つための「リアルモード」，80286の機能を生かす「スタンダードモード」，仮想86モード，仮想メモリが利用できる80386用の「386エンハンストモード」の3つのモードが1つのパッケージに収められており，Windows起動時にハードウェア環境を判断し，自動的に適正なモードで起ち上がる．

この新しいメモリ管理機構の採用によって，「スタンダードモード」では16MBまで，「386エンハンストモード」では4GBまでのプロテクトメモリを利用することが可能になった．日本語のパソコン環境では漢字を使うために，どうしてもメモリ領域が圧迫されるという問題があった．Windows 3.0の登場によって，ようやく解決するのではという期待がよせられている．



80386マシンで動作するWindows 3.0．「386 Enhanced mode」を表示し，仮想メモリを含むメモリ容量が示されている．

読者の中には，これからWindowsでのプログラミングを考えている人もいるだろう．Windows上でのアプリケーション開発のポイントについて，下川氏は次のように話してくれた．

「アメリカではビル・ゲイツが『PMやWindowsでのプログラミングは，慣れると生産性が向上する』といい，一方これに対する反論もずいぶんあるようです．でも，実際のところ慣れると作りやすいですね．ある目的のプログラムを作り上げる期間は，DOSの場合に比べるとかなり短縮されます．それに，設計するときもDOSの場合は処理を実現するための操作方法がいっぱいある．Windowsの場合は，『アプリケーションスタイルガイドブック』という本が用意されていて，メッセージはこうしなさい，メニュー展開はこうしなさいと書いてありますから，これにのっとって作ればいいので，設計であれこれ考えることもありません．そういう点はすごく楽です」．

「デバッグするときも，WindowsのAPIの1個1個がかなりの機能を含んでいますから，変なバグが出てこない．DOS版だとCで＊の数を1個間違えて，違うメモリ上にデータを書いてしまって動かないということがありますが，Windowsの場合は，違うメモリに書くということ自体がありません．周辺機器の操作についてもみんな関数にまかされていますから，わかりにくいバグっていうのは出てきませんね」．

「PMやWindowsのプログラミングの一番の問題は，WindowsとかPMの思想をいかに理解するか，ということだと思います．MS-DOSというのは，『周辺装置をよりソフトウェアからコントロールしやすくしよう』というレベルの思想しかない．でも，Windows, PMには明らかに思想がありますから，それをきちんと理解するということがまず一番目の仕事だと思います．これからWindows, PMに入る会社は，まず1年間はその勉強になるでしょうね．

OSの思想というのは，セミナーを聞いたり本を読んだりしてわかるというものではなくて，自分で体験しないといけません．そういう観点でものごとを捉えながら，いろんなサンプルソースを入れて動かしてみることが大切でしょう」．

PM, Windowsプログラミングのポイントは自分で体験してその思想を理解すること

PROFILE

25歳から飛び込んだプログラミングの世界 いまは「どんどん動いている仕事がおもしろい」



イースト株式会社
第2開発部第1開発課主任
藤井　徹治氏

ひところ「プログラマ30才定年説」が唱えられ，残業また残業のキツい仕事の代名詞であったプログラマ．ところが近ごろでは，知的(?)なイメージが幸いしたのか，プログラマをめざす学生も増えているという．実際にソフトウェア開発の最前線で働くプログラマの仕事ぶり，日常生活をさぐる．

パーソナリティ

年齢　1960年生まれの30歳.

趣味　音楽を聴くこと. 出歩くのはあまり好きではない.

趣味としてのパソコン　パソコンはあくまで仕事. 家には置いていない.

会社での仕事

開発環境とマシン　現在は, プログラミングにはEPSONの286を使用. Windows関係で三菱のMAXY, デバッグには, OKIのラップトップを使う.

エディタ　ほとんどMIFESしか使わない. 以前にWindows 2.1の仕事をしていたときは, ほかのエディタも使ったことはあるが, MIFESの方が好き.

開発体制　第2開発部第1開発課には主任が4人, それぞれ下に3人のプログラマがいる. はじめに「こんな話があるんだけど, やってみる？」と打診されて, そこで「やります」と答えると打ち合せに入ってプロジェクトが動き出す.

残業時間　去年は平均で月60～70時間くらい. 今年はそれほどでもない.

プログラマとしての経歴

入社して何年め？　2年2カ月.

プログラマ歴　プログラマになったのは25才のとき, 今年で丸5年になる.

プログラマという仕事　止まってることがなくてどんどん動いているのでおもしろい. 最近ではちょっとついていけないかなと思うこともある.

プログラミングの参考書　Kernighan&Ritchieの「プログラミング言語C」. しばらくしてから「はじめてのC」が出て, それがわかりやすかった.

これからやりたい仕事　特にない. 本当に仕事としか思っていないので, 来たらやらなくてはいけないというだけ. 3月の会議の時, これからどんな仕事をやりたいかたずねられたが, ひとりだけ「何でもいいです」と答えた.



藤井氏の机はごく普通の事務机. パソコンを何台も置くにはかなり窮屈そう. 実際, EPSON286の本体は机の下の段ボール箱の上に載っていた. タバコと灰皿がしっかりあるところを見ると, イーストのオフィスは禁煙ではないらしい.

プログラマの仕事場 | 2

Programmer's Working Space

平田祥司（株式会社マイクロフト）

制御系のハードウェア設計から その制御ソフトまでこなすから CADとCコンパイラが必需品

平田祥司氏の仕事は，ソフトウェアづくりといってもかなりハードからみのものが多い．たとえば，工場の生産ラインの制御系の仕事などは，ハードの回路設計にはじまり制御プログラムまで作ってしまう．回路設計もプログラミングもパソコン上で行うのだが，ソフト中心のプログラマがエディタ，コンパイラやツール類でほとんどの仕事ができるのに対して，平田氏はソフト開発のソフトと同じくらいの比率でCADを使う．CADで回路設計を行い，Cで制御プログラムを書く，これが平田氏のスタイルである．プログラムをROMに焼いてしまうだけに，ミスの許されないきびしい仕事である．

なぜ，平田氏はハードに強いプログラマになったのだろうか．彼のコンピュータとのつきあいを振り返ってみよう．

マイコンというものにはじめて接したのは，平田氏が高校生時代のこと．H68TRAという日立のワンボードマイコンを叔父に見せてもらったときだった．ゲルマニウムラジオからアマチュア無線へ，電気少年の道を歩いていた平田氏にとって，マイコンという機械はとても魅力的であった．部品を集めてワンボードマイコンをいくつか自作したあと，秋葉原の電気店でアルバイトをして，日立のベーシックマスターL3を手に入れた．

「ベーシックマスターL3は，キーボードと本体が一体になった大きな機械でしたが，その名のとおりBASICしか入っていなかったんです．そこで，アセンブラと逆アセンブラを自分で作って，それを使ってエディタを作り，もう少し高度なアセンブラと逆アセンブラを作り，コンパイラを何本も作りました」と，平田氏．「何かに役立つプログラムより，アセンブラのような開発用の環境を作るのが好きだったから……」という．

「学校にも行かずに暗いマイコン少年やってましたよ」と，平田氏は当時を思い出してたのしそうに話してくれる．「ぼろぼろの雑誌とテープレコーダーとカセットテープをもってアキバのBit-INNに行って，それをつないでゲームのプログラムを作り，またテープに落として帰ってくるんです」．いまでは，神話の部類に属するような行動パターンをもった典型的なマイコン少年だった．もっとも，現在のふくよかで笑顔のやさしい平田氏からは「暗さ」はまったく感じられないのだが．

「ぼくの場合，ほとんど独学でしたから，あまり本を見たという記憶はないですね．実際に機械をいじっておぼえていったわけですから，L3のROMと

カセットテープ片手にアキバへ 暗いマイコン少年やってました



Profile

平田祥司

1959年名古屋生まれ．

秋葉原　アマチュア無線が好きだった平田氏は，中学のときに東京に引っ越した．秋葉原に行けるのがとてもうれしかったとか．

GLP　パソコンネットではGFILEやGJOGなどのフリーソフトウェアの作者としても知られている．ファイルの整理などのために本人が必要に応じて作ったソフトだから，おなじようなソフトを望んでいた人から重宝がられる．

パソコン通信　現在は，それほどアクティブなネットワーカーではないが，パソコン通信で日本中に知り合いをつくった時期もあった．奥さんともパソコン通信で知りあった．

COLUMN

気分転換は別のプログラムかネットで

プログラムを書いていて，わけがわかんないけど，バグがとれないということはありますね．あとで見ると何でもないことなんだけど……．そんなときは，ちょっとそのソースから離れてみるといいんですが，ぼくの場合コンピュータが好きですから，同時進行している別のプログラムをいじることが多いです．そうでないときは，ほかの仕事の仕様書を読んでいたり，ネットに入ってメッセージ読んでいたり……．ネットは気分転換になりますね．

か，FM7のROMから教わったという面が強いです．アセンブラは，こうしたROMを逆アセンブルしておぼえたから……．ROMがぼくのお師匠さんというわけです．いまの人が，プログラミングをおぼえるには，ほかの人のソースを見るのがいちばん早いんじゃないですか」と話す．

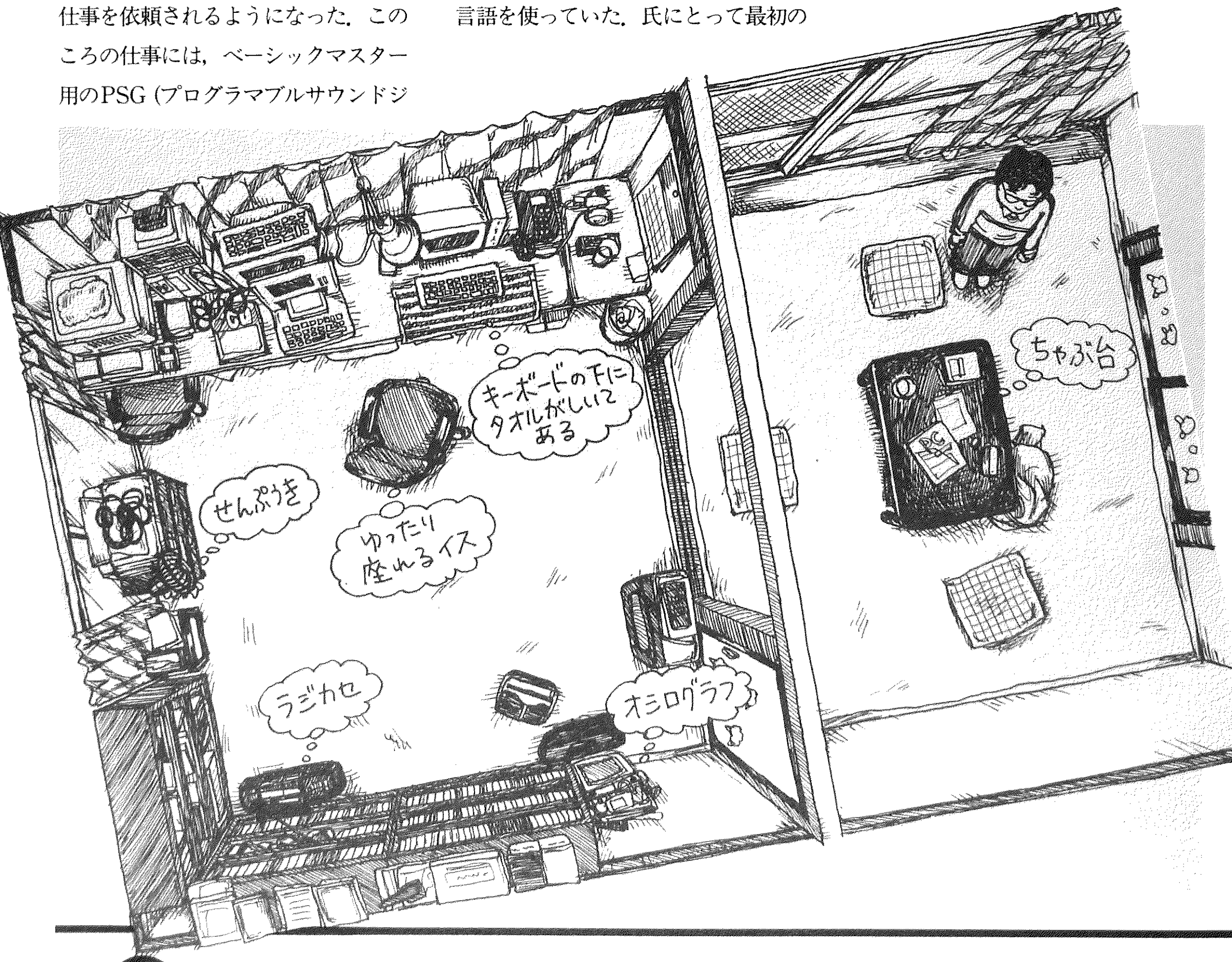
ハードがらみの仕事が多いのは それができるソフト屋さんが少ないため

✚

好きでマイコンをいじっていた平田氏に仕事がくるようになったのは，秋葉原のマイコン屋さんでアルバイトをしていたときのことだった．店によくくるメーカーの営業マンと知り合い，仕事を依頼されるようになった．このころの仕事には，ベーシックマスター用のPSG（プログラマブルサウンドジェネレーター）ボードの製作などがあった．雑誌『I/O』などで連載を持つようになったのもこのころのこと．大学1・2年のときだったという．

ROM化システム開発など，しだいに機械に近いプログラムの仕事が増えていった平田氏は，あまり日本語を使う必要がなかったため，ベーシックマスターL3の次に入手した，FM-77のFLEXやOS-9といったOS上でC言語を使っていた．氏にとって最初のPC-98であるVMを入手したのは，雑誌の原稿や翻訳の仕事が増えて日本語入力が必要になったからだった．以来，開発システムも98がメインになり，VX，RAとマシンを置き換えてきた．

現在は，制御系の仕事がメインで，制御用のボードの設計とそのソフトウェアを作ることと，既存のハードを使って制御プログラムを作ることが半々だという．「本当は会計や事務処理のアプリケーションをやってみたいとは思うんですが，ハードを扱えるソフト屋さんというのがあまりいないらしくて，制御系の仕事が多くなっ



ています」と，平田氏は話す．

✚

プログラムの仕事はほとんどTurbo-Cで行う．「8割くらいがTurbo-Cで，とくにお客さんからMS-Cを使うように要望されたときにMS-Cを使うくらいです」という．

今年になって，PC-9801VXをRA21に置き換えたため，開発環境はかなり強化された．CPUが80386になったのにともない，バンク切り替え式のメモリ4MBに加えて，プロテクトメモリを10MB増設した．「プログラムの仕事をしていると，エディタやコンパイラは1秒でも速く動いて欲しいです．コンパイルは何十回もかけるし，エディタはそれこそ何百回も起動しますから……．ほとんどRAMディスク上で仕事をします．ほんとうは，プロテクトメモリをRAMディスクにするといいんですが，プログラム開発中はよくハングアップしますから，リセットをかけると内容が消えてしまいますね．そこで，電源バックアップのあるバンク切り替え式のメモリをRAMディスクにあてています」．

エディタでもコンパイラでも 1秒でも速く動いてほしい

RAMディスクを使う理由は，まずエディタのロード・セーブが速くなること．ハードディスクからロードし，セーブするのとでは数秒の差がでることもある．ことに開発作業では，頻繁にロード・セーブを繰り返すことになり，RAMディスクによってかなりイライラを解消できる．次に，C言語では，RAMディスク上にヘッダファイルをすべてコピーすることによってコンパイルが速くなる．この点

COLUMN

RCS（リビジョンコンロトールシステム）

平田氏が開発用の便利なツールとしてあげたものにRCSがある．RCSは，リビジョンコントロールシステムの略で，ソースファイルのリビジョンを管理するプログラムである．作者は，平田氏の知人であるT.Tanaka氏だ．

プログラムのソースファイルは，改良されていく過程でいくつもの修正版（リビジョン）をもつことになる．たとえば，改良を重ねたソースファイルに重大なミスを発見し，何段階か前のリビジョンにもどらなければならないという状況になったとしよう．ところが，必要なソースファイルをすでに削除してしまっていたらどうだろうか．かなりの時間をむだにしなければならなくなる．また，すべてのリビジョンのソースファイルを保存しておくことも，多くのディスクスペースを消費することになるし，それぞれのファイルとその内容を管理すること自体なかなかたいへんなことだ．

RCSは，各リビジョン間の差分だけを記録して，ファイル容量をあまり大きくしないでソースファイルの遍歴を保存する．記録されたソースファイルは，任意のリビジョンを復元することが可能だ．さらに，アップデートの理由の記録，アップデートの履歴の参照，複数ユーザーによる同一リビジョンのアップデートを禁止するなど，ソースファイルを効率的に管理するためのさまざまな機能をもっている．

また，差分を記録してそれを表示できるため，プログラムのバグが見つけやすくなるというメリットがある．たとえば，昨日は問題がなかった箇所が今日の作業によってバグってしまったとしよう．そんなときは，昨日のソースと今日のソースの差分を表示することで問題のある部分を見つけやすくなる．

UNIXでは，このようなツールとして，SYSTEM VのSCCSとBDS版のRCSが知られているが，T.Tanaka氏のRCSは，UNIX BDS版のRCSがもっている機能をMS-DOS上で実現しようというもの．

RCSは，日経MIXなどのBBSで公開されており，掲載されているネットワークにアクセスすれば入手が可能だ．ただし，RCSは作者が寄付を受け付けている（とくに業務での使用）シェアウェア的なソフトなので，使用と配布については付属のドキュメントを参照されたい．

に関して平田氏は，コンパイラ本体やライブラリをRAMディスク上に置くなど，いろいろ試みてみたが，RAMディスクの容量とコンパイル速度の関係で，ヘッダファイルをRAMディスクに置くのがいちばん効率的だという．具体的には，ヘッダファイルの読み込みが速くなるため，コンパイラが最初のエラーメッセージをだすまでの時間が短縮される．↗

平田氏は，制御用のハードウェアの設計もしているが，それもPC-98上のCADソフトで行う．ソフトは，回路設計用CADとしてよく使われているOrCAD（注1）だ．

（注1）OrCAD/SDTIII（工人舎）
電子回路開発パーソナルCADと銘打たれた製品で，豊富なライブラリ，データベース機能をもつパーツリスト，結線もれをふせぐエラーチェック機能などが特徴．16万8000円．

表1：ハードウェア構成

本体	PC-9801RA21
ハードディスク	130MB
増設ドライブ	TS-MFD（3.5インチ2ドライブ）
増設メモリ	4MB（バンクメモリ）
	10MB（増設メモリ）
プリンタ	NM-9400，PC-PR102TL

図1：ディレクトリツリー

```
A:¥
|--BIN                              実行形式のファイルが主に入っている bin
|   |--DOS                          MS－DOSコマンド
|   |--EDITOR                       MIFESエディター
|   |--JEDIT                        日本語ワープロ（新松）
|   |--ORC                          回路設計用CAD（OrCAD）
|   |   |--BIN
|   |   |--DRV
|   |   |--TMP
|   |--PDS                          PDSが入っている
|   |--MSC                          MS－C
|   |   |--BINB
|   |   |--BINR
|   |   |--INCLUDE
|   |   |   |--SYS
|   |   |--LIB
|   |   |--SOURCE
|   |       |--DOC
|   |       |--ME
|   |       |   |--EXT
|   |       |   |--INI
|   |       |--SAMPLE
|   |       |--STARTUP
|   |           |--DOS
|   |--QC                           Quick C関係
|   |   |--BIN
|   |   |--INCLUDE
|   |   |   |--SYS
|   |   |--LEARN
|   |   |--SOURCE
|   |--TC                           Turbo－C
|   |   |--BIN
|   |   |--INCLUDE
|   |   |   |--SYS
|   |   |--LIB
|   |   |--PROGRAM
|   |--TERM                         通信ソフト（自作）
|   |--UT                           ツールなど
|--NET                              パソコン通信のLOGファイル
|   |--PDSARC                       PDSなどネットワークからダウンしたソフト
|--PCAKAIKE                         計処理ソフト（PCA会計）
|   |--PCARK
|   |--PCAZ1
|--SYS                              デバイスドライバなど
|--WK1                              作業用のディレクトリ
|--WK2
|--WK3
|   |--CBASE4
|   |   |--H
|   |--LIBS
|   |   |--RCS
|   |--TMP
|--WK4
|   |--DEV
|   |--RCS
|   |--SAMPLES
|--WK5
|   |--H
|   |   |--LATTICE
|   |   |--MSC
|   |   |--NET
|   |   |--NETINET
|   |   |--SYS
|   |--LATTICE.LIB
|   |--MSCLIB
|   |--SAMPLE.SRC
|   |   |--LAT.S
|   |   |--MSC.S
|   |--TESTLAT.EXE
|   |--TESTMSC.EXE
|--XLN                              ネットワーク関連開発ツール
    |--BIN
    |--HARDWARE
    |--HSTACC
    |--SAMPLE
    |--TCP
```

CAD
C COMPILER

「プログラムの仕事は，いくつもが同時進行することが多いんです．そこで，ソースが混ってしまうとたいへんなので，PKZIPでひとつの仕事に1個のZIPファイルというようにソースファイルをまとめながら仕事をしています」と話すように，プログラムのソースファイルは，毎日の仕事によってかなりの分量になる．それを，そのままハードディスクの作業ディレクトリに収めておいたのではもったいない．そこで，ファイル圧縮プログラムを用いて，ソースフ↙ァイルを保存するようにするとハードディスクをより効率的に利用できることになる．PKZIPとは，ファイルの圧縮を行うシェアウェアだ．PKZIP.EXEで圧縮したプログラムは，PKUNZIP.EXEによって復元する．

1つの仕事はZIPファイルにまとめ RAMに展開してから作業する

RAMディスク上ではほとんどの作業を行うから，ファイルを普通に扱える状態にしておくのはそのときだけでいい．ハードディスクには，いつも圧縮ファイルだけが保存されていることになる．これだと，RAMディスクで作業をしていて変更してはなら

ハードから設計することが多いので共通ライブラリはあまり使えない

ないファイルを編集してしまっても，すぐにハードディスクから元のファイルを取り出すことができる．平田氏の作業を説明したのが図2である．

RAMディスクのファイルをハードディスクに圧縮して書きもどすときは，PKZIPの起動時にパラメータをつけることによって，更新されていないファイルをオーバーライトするむだを省くように設定している．RAM上であるだけに，作業の切れ目にはディスクに書き込んでファイルを更新するようにしている．そのためのBATファイルがDONE.BATで，作業の切れ目，そして1日の仕事が終わったときに，DONE.BATを実行している．

またハードディスクは，ディレクトリにわけてプログラムやデータを管理するのに適した環境であるが，使うプログラムによってパスを変更したりするのは面倒なものだ．それを簡単に行うためにG.BATというBATファイルを作り，仕事をはじめるときはそれを実行するだけで環境が作られるようにしている．

たとえば，

```
A> G TC2
```

と入力するとTurbo Cを動作させるのに必要なパスが通り，プログラムを作る環境が設定される．

仕事が動いているうちは，ハードディスクの圧縮ファイルをフロッピーにバックアップする．ZIPファイルひとつだけだから，1日にいくつかの仕事をしてもバックアップは簡単だ．そして，ひとつの仕事をすべて終えると，バックアップしたフロッピーを銀行の貸金庫におさめる．こうしておけば，「たとえ仕事場が火事になってもソースだけは大丈夫です．なにかあってもハードは買えばいいけど，ソースは取り返しがつかないですから」というわけである．

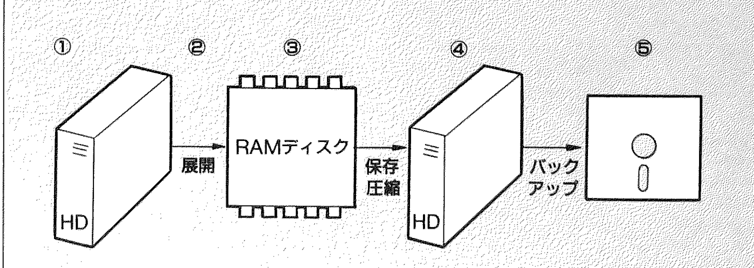
✚

「ライブラリは，けっこう作っているほうでしょう」と，平田氏は話す．PC-98のRS-232Cまわりのもの，画面にウィンドウを出したり閉じたりというもの，昔はTurbo-Cにグラフィックライブラリがついていなかったため，それも作った．また，数字の列にカンマを入れるような関数など変換用ライブラリも用意してある．MYLIBS.LIBというライブラリにしてインクルードしている．しかし，制御系のシステムは，ハードが共通であることはほとんどないため，共通ライブラリを使えるような仕事はあまりない．

むしろ，ハードの関係のツールが揃っている．高速なROMライターやエミュレータのほうが平田氏の仕事には役立つのだという．

図2：平田氏の作業の流れ

①ソースファイルはハードディスクに圧縮されて保存してある
②圧縮ファイルをPKUNZIPでRAMディスクに展開する
③RAMディスク上で作業をする
④RAMディスクから更新したファイルを圧縮してハードディスクにコピーする
⑤ハードディスクのファイルをフロッピーにコピーする



リスト1 DONE.BAT

```
set BFILE=a:¥wk3¥x25
IF "%1"=="-F" GOTO D0
IF "%1"=="-f" GOTO D0
GOTO D1
:D0
DEL %BFILE%*.*
:D1
del *.BIN
del *.MAP
del *.BAK
del *.obj
pkzip -prues %BFILE% *.*
```

リスト2　G.BATの内容

```
echo off
set a=
set a1=
set a2=

PATH %HD%¥BIN¥PDS;%HD%¥;%HD%¥BIN¥dos;%HD%¥BIN¥UT;%HD%¥BIN¥editor;%tmp%;a:¥xln¥bi
n;
rem   20 19  18 17 16 15 14 13 12 11 10  9  8  7
CASE @%1 tc2 jp orc db lotus  5 hd  d tc qc introl hc fax  msc
if ERRORLEVEL  20 goto TURBOC2
if ERRORLEVEL  19 goto JPEDITOR
if ERRORLEVEL  18 goto ORCAD
if ERRORLEVEL  17 goto DBASE
if ERRORLEVEL  16 goto LOTUS
if ERRORLEVEL  15 goto BOOTFD
if ERRORLEVEL  14 goto HD
if ERRORLEVEL  13 goto CATCH
if ERRORLEVEL  12 goto TURBOC
if ERRORLEVEL  11 goto QC
if ERRORLEVEL  10 goto INTROL
if ERRORLEVEL  08 goto FAXINIT
if ERRORLEVEL  07 goto MSC

echo ********************************************************
echo   [43m  =====  SYS.BAT COMMAND ERROR !!  =====   [37m
echo ********************************************************
VBELL -V 30 30 30 30
ECHO                     H E L P    T A B L E
ECHO ********************************************************
ECHO  G INTROL    : INTROL C SETUP
ECHO  G TC        : TURBO-C 1.5 SET UP
ECHO  G TC2       : TURBO-C 2.0 SET UP
ECHO  G ZTC       : ZORTECH C SET UP
ECHO  G TERM      : GTERM SET UP
ECHO  G DB        : DBASE-III SET UP
ECHO  G LOTUS     : LOTUS 1-2-3 SET UP
ECHO  G JP        : JAPANESE EDITOR SET UP
ECHO  G D         : CHATCH
ECHO  G ORC       : OrCAD
ECHO  G 5         : BOOT FROM 5" DISK
ECHO  G HD        : BOOT FROM HD
SWITCH BOOT[SHD] MEMORY[640] RS0[9600 B8 S1 PN] ndp1[yes]
quit

:TURBOC
echo ********************************************************
echo                     TURBO-C
echo ********************************************************
echo on
PATH %HD%¥BIN¥TURBO¥BIN;%PATH%
cd %HD%¥bin¥turbo
MD %tmp%¥TC
%tmp%
CD %tmp%¥TC
if "%2" == "" GOTO TC00
F:
fmark a:¥sys¥thelp
thelp
release a:¥sys¥tc
fmark a:¥sys¥tc
%tmp%
:TC00
quit

:TURBOC2
echo ********************************************************
echo                     TURBO-C 2.0
echo ********************************************************
echo on
PATH %HD%¥BIN¥Tc¥BIN;%HD%¥bin¥td;%PATH%
%HD%
md %tmp%¥tc
md %tmp%¥thelp
cd %HD%¥bin¥tc
THELP /D%tmp%¥thelp /Fa:¥bin¥tc¥bin¥tchelp.tch /K003f /C170 /C350 /C550 /C761 /S
1
release a:¥sys¥tc
fmark a:¥sys¥tc
%tmp%
CD %tmp%¥TC
quit

:MSC
echo ********************************************************
echo                     MS-C 5.1
echo ********************************************************
echo on
PATH %HD%¥BIN¥MSC¥BINR;%HD%¥BIN¥MSC¥BINB;%PATH%
set LIB=A:¥BIN¥MSC¥LIB;%LIB%
set INCLUDE=A:¥BIN¥MSC¥INCLUDE;%INCLUDE%
set TMP=%TMP%
set INIT=A:¥BIN¥MSC¥SOURCE¥ME¥INI;%INIT%
md %tmp%¥MSC
cd %HD%¥bin¥MSC
release a:¥sys¥MSC
fmark a:¥sys¥MSC
CD %tmp%¥MSC
%tmp%
quit

:QC
echo ********************************************************
echo                     QC
echo ********************************************************
echo on
PATH %HD%¥BIN¥QC¥BIN;%PATH%
set LIB=A:¥BIN¥qc¥LIB;%LIB%
set INCLUDE=A:¥BIN¥qc¥INCLUDE;%INCLUDE%
set TMP=%TMP%
set INIT=A:¥BIN¥qc¥SOURCE¥ME¥INI;%INIT%
md %tmp%¥MSC
cd %HD%¥bin¥MSC
release a:¥sys¥MSC
fmark a:¥sys¥MSC
CD %tmp%¥MSC
%tmp%
quit
:INTROL
echo ********************************************************
echo                     INTROL-C
echo ********************************************************
echo off
IF EXIST %tmp%¥USR¥INTROL¥BIN¥CO.EXE GOTO SUP
:ALLINTROL
echo ********************************************************
echo *   ｚｉｐファイル集をドライブＤに入れてください。   *
echo ********************************************************
pause
MD %tmp%¥usr
pkunzip -d d:INTROL %tmp%¥USR
%tmp%
CD %tmp%¥USR
MD %tmp%¥USR¥INTROL¥WORK
MD %tmp%¥USR¥INTROL¥INCL09
COPY %tmp%¥USR¥INTROL¥INCLUDE¥STDIO.H %tmp%¥USR¥INTROL¥INCL09
```

```
:SUP
PATH %tmp%¥USR¥INTROL¥BIN;%PATH%
cd %HD%¥;
%tmp%
cd %tmp%¥usr¥introl¥work
quit

:ZTC
echo ********************************************************
echo                    ZTC
echo ********************************************************
echo on
SET PATH=%HD%¥BIN¥ZORTECH¥BIN;%HD%¥bin¥td;%path%;
SET LIB=%HD%¥BIN¥ZORTECH¥LIB
SET INCLUDE=%HD%¥BIN¥ZORTECH¥INCLUDE
cd %HD%¥bin¥zortech
md %tmp%¥ztc;
%tmp%
cd ¥ztc;
quit
:DBASE
echo ********************************************************
echo                    dBASE III pluse
echo ********************************************************
echo on
%HD%
cd %HD%¥bin¥dbplus
DBASE
quit

:LOTUS
echo ********************************************************
echo                    LOTUS
echo ********************************************************
echo on
%hd%
CD %hd%¥bin¥lotus
LOTUS
quit

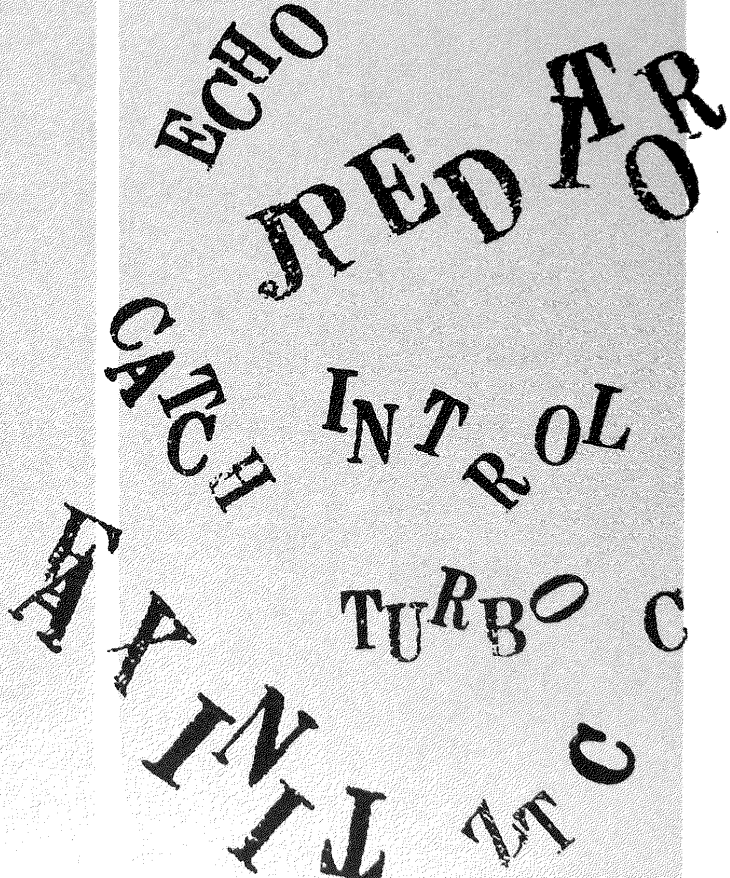
:BOOTFD
SWITCH BOOT[STD] MEMORY[640] RO[9600 B8 S1 PN]
BOOT

:HD
SWITCH BOOT[SHD] MEMORY[640] RO[9600 B8 S1 PN]
quit

:CATCH
echo ********************************************************
echo                    CATCH
echo ********************************************************
a:
CD A:¥BIN¥CATCH
CATCH
quit

echo on
:JPEDITOR
echo ********************************************************
echo                    JAPANESE EDITOR
echo ********************************************************
echo on
%hd%
cd %hd%¥bin¥jedit
shift
matu
quit
echo on
:ORCAD
echo ********************************************************
echo                    ORCAD
echo ********************************************************
echo on
palinit
%HD%
cd %HD%¥bin¥ORC¥BIN
DRAFT %2 %3 %4
quit

:FAXINIT
echo ********************************************************
echo                    FAXINIT
echo ********************************************************
echo on
ECHO      > PRN
MKDATE "+$$$HALFAX TIME %%y/%%m/%%d,%%H:%%M"     >prn
ECHO $$$HALFAX SETID  03-XXX-XXXX                >prn
ECHO $$$HALFAX SETHID "㈱マイクロフト"            >prn
ECHO $$$HALFAX NORMAL                            >prn
ECHO $$$HALFAX CLRSBF                            >prn
ECHO $$$HALFAX CLRRBF                            >prn
ECHO $$$HALFAX CLRPBF                            >prn
quit
```



プログラマの仕事場 | 3

Programmer's Working Space

マイクロソフト株式会社

業界トップの開発環境は国際ネットワークに接続されたハイパワーシステム

パソコンの標準的なソフトウェア環境であるMS-DOS，そしてアメリカでの強烈なデビューの余波がまだおさまらないWindows 3.0など，マイクロソフト社の製品は，常にユーザーからも開発者からも注目されてきた．それだけに，同社の開発環境がどういうものであるかプログラマならずとも興味の的であろう．
ここでは，同社創立以来の開発スタッフである増川美樹氏（研究開発部アプリケーション開発課主事）に登場していただいた．

ご存知のようにマイクロソフトの本社は米国ワシントン州にあり，マイクロソフト社から日本市場に提供されるOSやアプリケーションなどの日本語版は，すべて米国のオリジナルをもとに，日本のマイクロソフトで日本語化の作業が行われている．したがって，開発環境の特色としては，米国本社との共同作業が多い点があげられるだろう．増川氏が最近関わった，統合ソフトMicrosoft Worksの日本語化作業を例に，日米共同によるアプリケーション開発の様子を見てみよう．

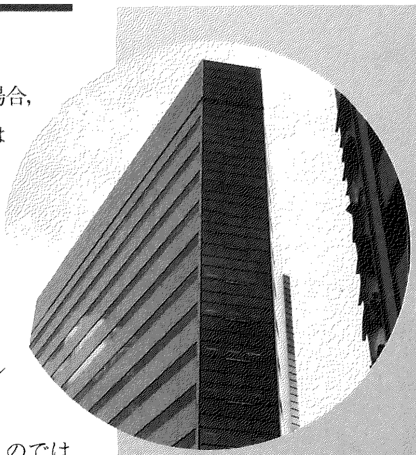
以前は，米国版が発売されてから改めて日本語化が検討され，日本語化プロジェクトがスタートする場合も見受けられたが，最近では米国での開発当初から日本語化が想定され，同時進行で開発が進められるケースも多くなってきた．このWorksも日米共同で開発が進められたアプリケーションだという．

「先行として米国版のPC-Worksが動いていたんですが，ほぼ同時に日本語の機能を組み込んだかたちで進行しました．普通ですと米国版に比べて半年から1年遅れで日本側のプロジェクトは立ち上がるんですが，Worksについてはほぼ同時進行で行ったわけです．最終的には発売時期が漢字Worksのほうが半年ぐらいずれましたけれど，いつものような進行状況でやっていたらもっと遅くなっていたかも知れませんね」と，増川氏．

日米の共同プロジェクトの場合，日本国内だけで開発するのとはずいぶん環境も違うし，勝手も違ってくる．電子メールのやりとりだけでは意思疎通が図れないことも多く，また時差の関係で昼夜が逆になってしまうため，ディスカッションもままならない．

「ぼくは直接関わったというのではないんですが，今年の3月くらいまではWorksの画面用ドライバなどを書くなど間接的には関わっていました．それで何度かアメリカのほうに出張しましたが，精神的にも肉体的にも疲れますし，たいへんですね．それと言葉の問題があるので，細かなニュアンスがどうしても伝わらないという悩みがあります．大まかなことを伝えるんでも苦労するんですから，細かなところとなるとなおさらです．それで現在は，アメリカに日本から1人，専任として常駐してもらって，あとはスポット的に2～3か月単位でこちらから出張するという形で進めています．それでも日米間の距離がありますから，難しい面もありました．次のプロジェクトでは分担という形ではなく，日本で全部やるということを前提に，アメリカから何人かのヘルパーにきてもらうということも考えています．そうすれば，あとは日米でメールのやりとりを

日米で同時進行したWorksの開発 共同プロジェクトは連携が難しい

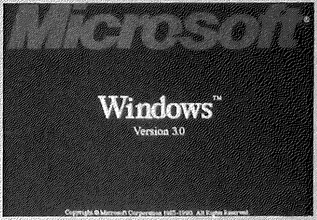


Location

新宿駅に近い高層ビルにオフィスを構えるマイクロソフト社．眼下に新宿の町が広がり，晴れた日には多摩の山々も青い影を映す．そこには，サイバーパンクシティの様相を呈している新都心とは異なる別世界がある．

About Company

マイクロソフト株式会社，古川享社長．マイクロソフト社は，MS-DOS，Windows，OS/2などのパソコンOS環境，Multiplan，Excelなどのアプリケーション，そしてプログラマには欠かせないMS-Cなどで，現在のパソコンの世界をリードしてきた会社だ．その研究開発部門である研究開発部は総勢112名の大所帯．米国本社との連係の下，研究開発に取り組んでいる．



Microsoft社

1975年に，はじめてマイコン用BASICを開発したビル・ゲイツとポール・アレンが創立したソフトウェア会社．あらためて書くまでもないが，MS-DOS，MS-Win



する範囲で済むとも思うんです．何度かこちらのチームがアメリカへ行って打ち合せする程度のことは必要でしょうが」．

✚

さて，マイクロソフトともなれば，開発にどんな言語やツールを使っているのか興味のあるところだ．特別な秘密兵器があるのだろうか．

「それがぜんぜん特別なものはなくて，当然うちのプロダクトとして発売されているMS-Cとアセンブラですよ．バージョンは現時点では両方とも5.1です．市販品とちょっと違うという点は，社内で使いながら若干の改良を加えていきますから，厳密にいえばバージョン5.1の下のリビジョンは変わっていることになります．こういう日々の改良がまた製品のバージョンアップとなってユーザーに還元されていくというわけです」と，増川氏は語る．自社の開発現場で製品を使い込み，それが製品を磨きあげることにもつながるわけだ．

基本的には自社製品で開発するが社内用の開発ツールも使われている

とくに社内だけで使っているものとして，普通のCコンパイラだけではなく，内部用のPコードをはき出すPコンパイラがある．これは，最初に発売されたMultiplanの時代から使われている．

このPコードは一種の中間コードで，普通のコンパイラだとマシン語まで直接落ちるが，Pコードではそこまで落とさず，中間に仮想のマシンコードのようなものを定義して，実際に動くときにはインタープリタがそれを解釈しながら動かすという手法である．

こうしておくと，インタープリタ自体は小さなものになるため，各マシンにインタープリタだけを移植すれば，オブジェクトレベルで動作環境間の互換性がとれることになる．したがって，ExcelなどではPコードでPC/ATにも，マッキントッシュにものり，ほとんど同じオブジェクトコードで動かすことができる．最終的には同じもの↙にはならないが，かなり共通な部分が残せるという利点があるために使われているようだ．

「しかしPコードも，コンパイラの質が悪かった時代にはかなり有効な手段だったようですが，いまの現状ではどこまで有効性があるかどうか．ただ使い慣れていますし，信頼性も高いということで使っています．今後はだんだんなくなっていくんじゃないかと思います．オブジェクト指向のC++のような方向になっていくでしょうね．最新のMS-Cはバージョン6.0ですが，いずれC++に準拠したコンパイラになるでしょう」．

✚

マイクロソフト社は，社内そして米国，さらにはヨーロッパをネットワークで接続している．ネットワークは図1のような構成になる．米国本社とはリモートブリッジを介してデジタルラインでつながれている．社内のLANは，幹線を通じて各セグメントにつながれており，マシンルームには40～50台のサーバーがおかれている．

それぞれのデスクにおかれている開発用マシンは，コンパイルやエディット用にはおもにコンパックが使われている．そして，ターゲットマシンとしてAXや98があるという形である．各人最低で2台，コンパック+AX，あるいはコンパック+98の組み合わせで

使っている．人によってはデスク上に3台，4台のマシンがおかれることになる．これらのマシンはネットワークでつながれており，となりのマシンにデータを移すのもネットワーク経由で行ってしまうという．

社内の平均的なところでは，マシンはコンパックの20～25MHz，110MB程度のハードディスクを積むという構成のようだ．また，アプリケーションのチームは現在，全員がメモリを10MB持っているため，使いたい人はOS/2を使うということにしているそうである．最近はOSにOS/2を使う人が増えてきたという．

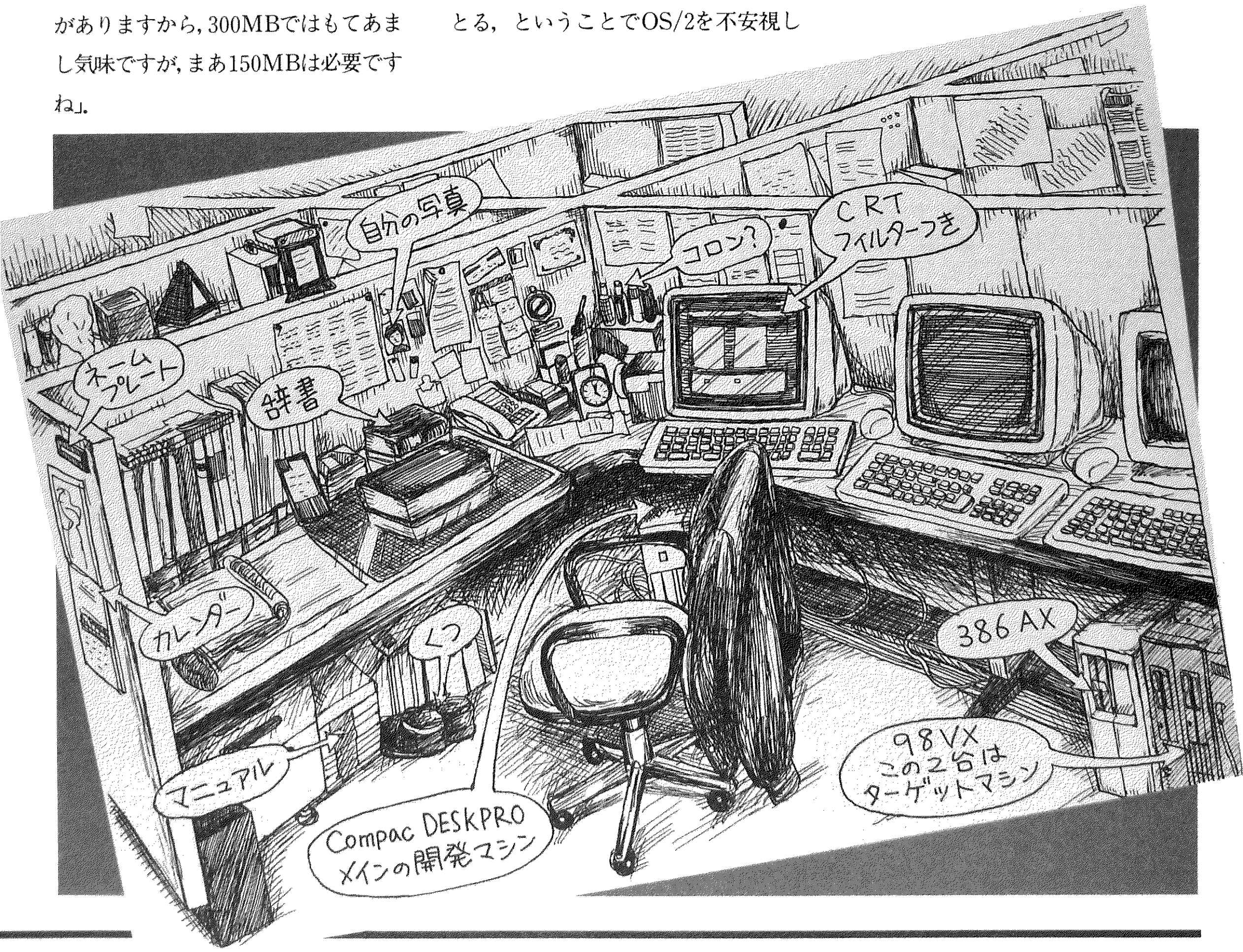
「はじめのころはメモリもたくさんとる，ということでOS/2を不安視し

コンパックDESK PRO+10MB以上のメモリが標準 ネットワークは欧米に接続している

「ぼく自身はコンパックDESK PROの33MHzを使っています．OSはOS/2で，メモリ16MB，ハードディスクは300MBという構成です．サーバーがありますから，300MBではもてあまし気味ですが，まあ150MBは必要ですね」．

dowsなどのMSとはマイクロソフトを意味している．パソコン用の開発・動作環境としてBASICが普及したのち，IBM PCが世に出て，PC-DOSが標準的なオペレーティングシステムになるとともに，PC-DOS＝MS-DOSが世界のパソコン界を席巻した．現在も，マルチタスクのOS/2，MS-DOSのグラフィックベースOS環境Windowsなど，システムソフトウェアでは他をよせつけない地位を確保している．

コンパイラなど開発用ソフトウェアでも，その製品への信頼は高い．プロ用Cコンパイラの標準的存在であるMS-C，MASM，QuickBASIC，QuickCなどをはじめとする製品群は，広い範囲をカバーしている．また，Excel，Multiplan，Worksなどのアプリケーションソフトメーカーとしても重要な存在である．



ている人も多かったんですが，メモリが安く手に入るようになったのでOS/2に切り換えています．本社ではほとんどOS/2なんで，本社からくるツール類がOS/2でしか動かないという環境もあって，こちらでもそれにならうかっこうで切り換えられてきているわけです．OS/2はメモリ関係が強力ですからね．開発環境の場合，裏でコンパイルさせられるのは快適ですね」．OS/2のマルチタスク環境での開発は，使いはじめるともう元にはもどれないと，多くのプログラマから耳にする話である．

パソコンの自由さを守りながらセキュリティを保つことの難しさがある

✚

コンパイラやツール類は，すべてネットワークのあるサーバーの中に収められており，各自が必要に応じてサーバーから引き出して使っている．サーバーのほうのディスクは，各サーバーごとにそれぞれのプロダクトだけを集めてあり，ソースだけのサーバー，ツール類を集めてあるサーバー，あるいはプリンタ用のサーバーなどに分類してあるという．

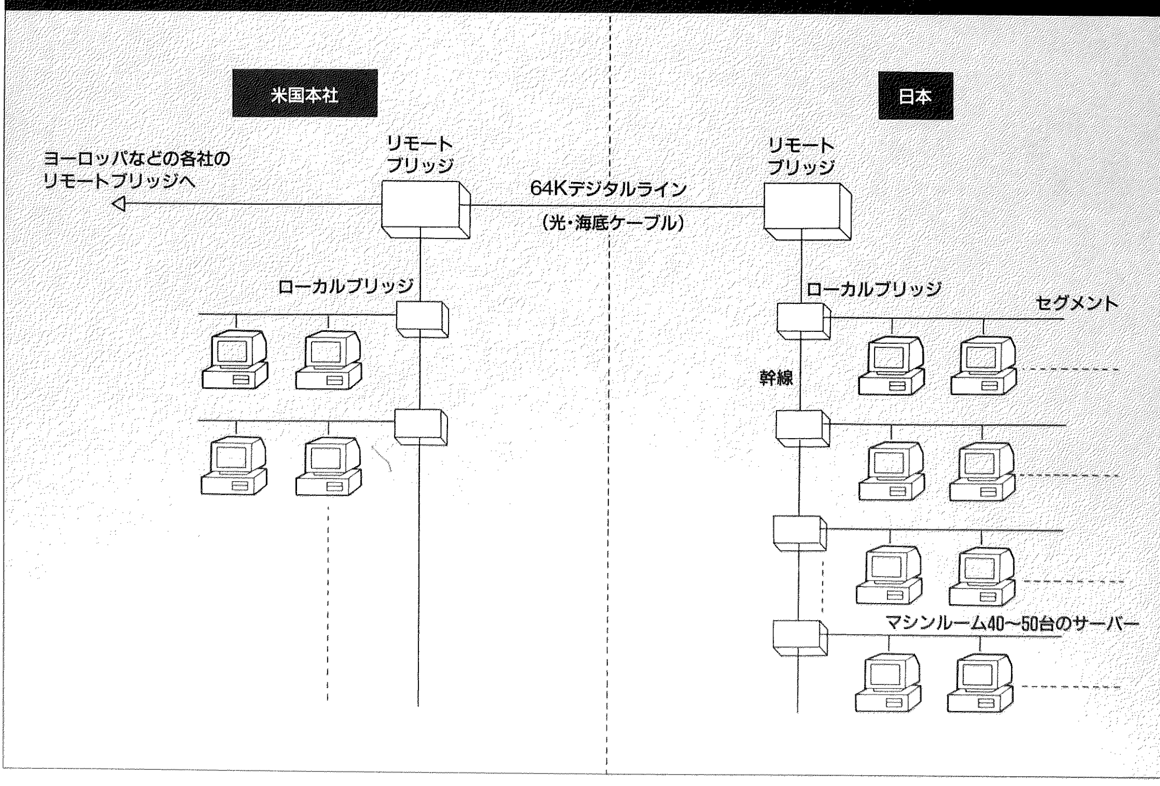
ツール類の中でもエディタは個人の好みがでるものだが，増川さんの愛用は何だろうか．

「ぼくはずっとviを使ってますね．最近はMIFESなどを使っている人もかなり多くなっていますが，viがいま半分ぐらいでしょうね．UNIXを知っている人たちはviを使う，それを知らない人たちはMIFESなどになっているんじゃないでしょうか．本社でもviは減っているみたいで，ブリーフというのがかなり使われていますね．日本語版のブリーフはOS/2 PM用というのがないようですが，あったら使ってみたいですね」．

ネットワークのセキュリティ関連について，おおまかなところを増川氏に語ってもらった．

「MS-NetworkやLAN Manag-

図1：マイクロソフト社のLANシステム



PROFILE



増川美樹

年齢　30歳.

パソコンとの出会い　高校２年のころ，趣味にしていた天文学に出てくる面倒な軌道計算をマイコンでできないものかと興味をもったのがきっかけ．ワンボードマイコンが出始めたときで，マイコンショップに出入りしはじめた．

ひょんなことから　『月刊アスキー』編集部でアルバイトをするようになり，以来アスキーの開発部門を経てマイクロソフト設立とともに移籍，現在まで開発部門の第一線に．

OS畑　BASIC, CP/M, MS-DOSとOS関連の開発を歩き続けてきたが，２年前からアプリケーション部門へ転じた．「OS/2から先はもっと若い優秀な人たちに任せようと思いまして」とは本人の弁．

アスキー時代　「アスキー時代に古川社長と知り合って，編集部時代から一緒で，研究部門に移ったときも彼が責任者でしたからずっと一緒でした．その縁で，マイクロソフト設立の時にも一緒にきたようなわけです．アスキー時代には古川さんもCP/Mのコードとか書いていたんですよ．古川さんはすごくできましたよ．ディスクのコントローラのコードとかをスラスラ書いてましたから」

erのセキュリティレベルでディスクアクセスするときは，必ずパスワードが必要なようになっています．ですからそのパスワードを知らない人は，そのサーバーはアクセスできません．例外として，みんながアクセスしてもいいようなところは，パスワードなしでアクセスできます．それはPDSなどを入れてあるようなサーバーに限ってですが．

現在は，まだMS-Networkのレベルのものが半分くらいあるので，セキュリティレベルが低いんですが，順次LAN Managerのサーバーにかえていく予定ですので，そうなるとパスワードだけではなくて，たとえパスワードを知っていても特定の人でなければ受付けないような体制になりますね．パソコンですから，あまりにセキュリティでガチガチにしてしまうと使いづらくなりますし，パソコンの自由さを守りながらセキュリティを考えるということは難しいことですよね」．

製品の機能を企画することより
その機能を実現していくことが楽しみ

30歳の若さながら，パソコンプログラマとしては長年のキャリアをもつ増川氏．最後に，氏のプログラマ観といったものを聞いてみた．

「開発部門には企画系とプログラム系とありますが，ぼくはプログラム系のほうがいいですね，プログラムを書いているほうが気が楽ですから．自分の知識を総動員して問題を解決していくということがじっくりとできますから．企画の人は，こういう機能を盛り込もうというところで終わるわけですが，プログラマはそのあと，その機能をどう実現するかということを考えていくわけですから，作り上げる楽しさはあると思います．

プログラマの30歳とか35歳限界説なんてのもありますが，アメリカの本社では40歳や45歳ぐらいのプログラマはたくさんいるんで，まあぼくらもできるんじゃないでしょうか．ただし，常に頭を使っていないとすぐ固くなりますから，その点は注意しないと...」．

プログラマの仕事場 | 4

PW
Programmer's Working Space

西川和久（フリープログラマ）

より広いコンピュータの世界と快適な開発環境を求め続けるアクティブなプログラマの生活とは

経歴からみると，西川氏は純粋な「パソコン少年」であったといえる．小学校のときから機械のおもしろさにとりつかれ，ラジオや管球アンプを組み立て，マイコンが登場するや参考書を片手にたちまち自作マイコンを組み上げてしまう……．以来，コンピュータから離れることはなかった．中学，高校，大学と10代をつうじてコンピュータとのつきあいをつづけ，現在はプログラミングの仕事を続けている．

こう書いてしまうと，読者はいかにも内向的なパソコン少年の成長した姿を思い浮かべるかも知れないが，実像はまるで違う．コンピュータにかかわる情報への幅広い興味と知識，自らが快適だと感じるハードウェア・ソフトウェア環境を選び築きあげる実行力，西川氏にはいまいちばんアクティブなプログラマ像を見ることができる．

オンラインソフト情報
INIT.SYS(INIT203.LZH)　DOS起動時にメニューで各デバイスドライバを選択にするデバイスドライバ

ハードウェアがおもしろかったから コンピュータにはまってしまった

マイコンが登場したとき，まっさきにそれに注目したのは，機械好きのラジオ少年たちとコンピュータに知識がある研究者たちだったろう．いずれの場合も，マイクロプロセッサという新技術に夢を託したという共通点があった．

ラジオやオーディオアンプを作るのが好きだった西川氏が，マイコンキットの存在を知ったのは中学２年のときだった．1976年の夏，日本電気からマイコントレーニングキットTK-80（注1）が発売された．米国では，スティーブン・ジョブズのガレージでアップル社が誕生した年だ．



このマイコンキットは，すぐにでも手に入れたかったが，中学生にとって「とても自前では買えなかった」から，「お年玉を貯めて」参考書片手に自作することにした．このときに使った参考書は『私だけのマイコン設計&製作』（CQ出版）であった．この本には，CPUに6800を使ってマイコンを製作する記事が掲載されていたが，西川氏はCPUを6802にかえて部分的に変更を加えた回路を組んだのだった．半田ペーストのシミが残る古びたその本はまだ彼の部屋に残っている．

6802をCPUとし，512バイトのメモリで自作したマイコンは，その後メモリを１KBに増設し，CRTとキーボードもつけられた立派なマイコンになった．

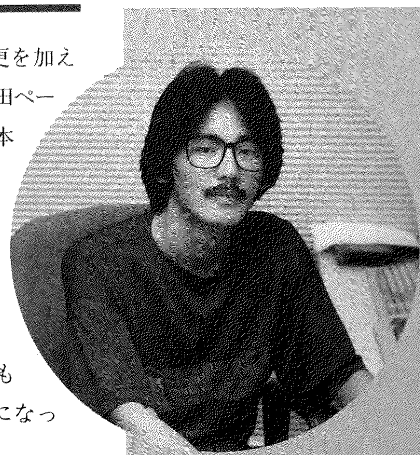
「もともとハードウェアが好きだったんですよ．コンピュータを使うためにハードをいじったんでなしに，ハードウェアがおもしろかったからコンピュータをさわるようになったんです」という．

✚

大学に入ってからシャープのMZ-80K2Eを買った．製品としてはじめて買ったパソコンであった．それから，機種は，PC-8800，PC-9800シリーズへとかわっていったが，「自前で買ったのはMZが最後だったと思います」という．大学の在学中にプログラムの仕事をすることになり，それからは仕事に応じてさまざまなマシンにふれることになったからだ．

最初の仕事は大学２回生のとき，カ

（注１）TK-80
CPUにμPD8080AD（2MHz）を搭載したワンボードマイコンキット．1976年８月に日本電気から発売された（８万8500円）．入出力に８桁のLEDと16進キーボードがついただけのものであったが，マイコンマニアたちは，これを手に入れるために秋葉原に殺到した．



Profile

西川和久
1962年　奈良県で生まれた．ワンボードマイコン時代からコンピュータにかかわった世代では最も若いほうに属する．
大阪を拠点として　大学時代からプログラムの仕事をしてきた．1989年に東京に移転．1990年にはCFコンピューティングを設立する．
CFC　BBSで公開しているCFCというプログラムは西川氏の作品．AXパソコンの日本語モードで米国のソフトを起動したときに，罫線が半角カタカナなどに化ける現象を解決する常駐プログラムだ．



CFCを使わないと



CFCを使うと

シオの家庭用ゲーム機のゲームプログラムを組むものだった．そのときにははじめてCP/M上でプログラムを書いた．

ゲームは，そのときに2本書いただけで，次にはワンボードのコントローラーを作る仕事を受けた．CPUボードの回路自体を設計することからはじめて，2KBのROMに入れる制御プログラムまで作った．

はじめての仕事はゲームのプログラム
そのままプログラムの世界に入ってしまった

そのころ，仕事はひょんなところからくることが多かったという．

当時，UNIXマシンに興味をもっていた西川氏は，大阪でただ1軒UNIXマシンを展示してあったコムスポットという店に通いつめていた．やがて，店の人と仲良くなった．

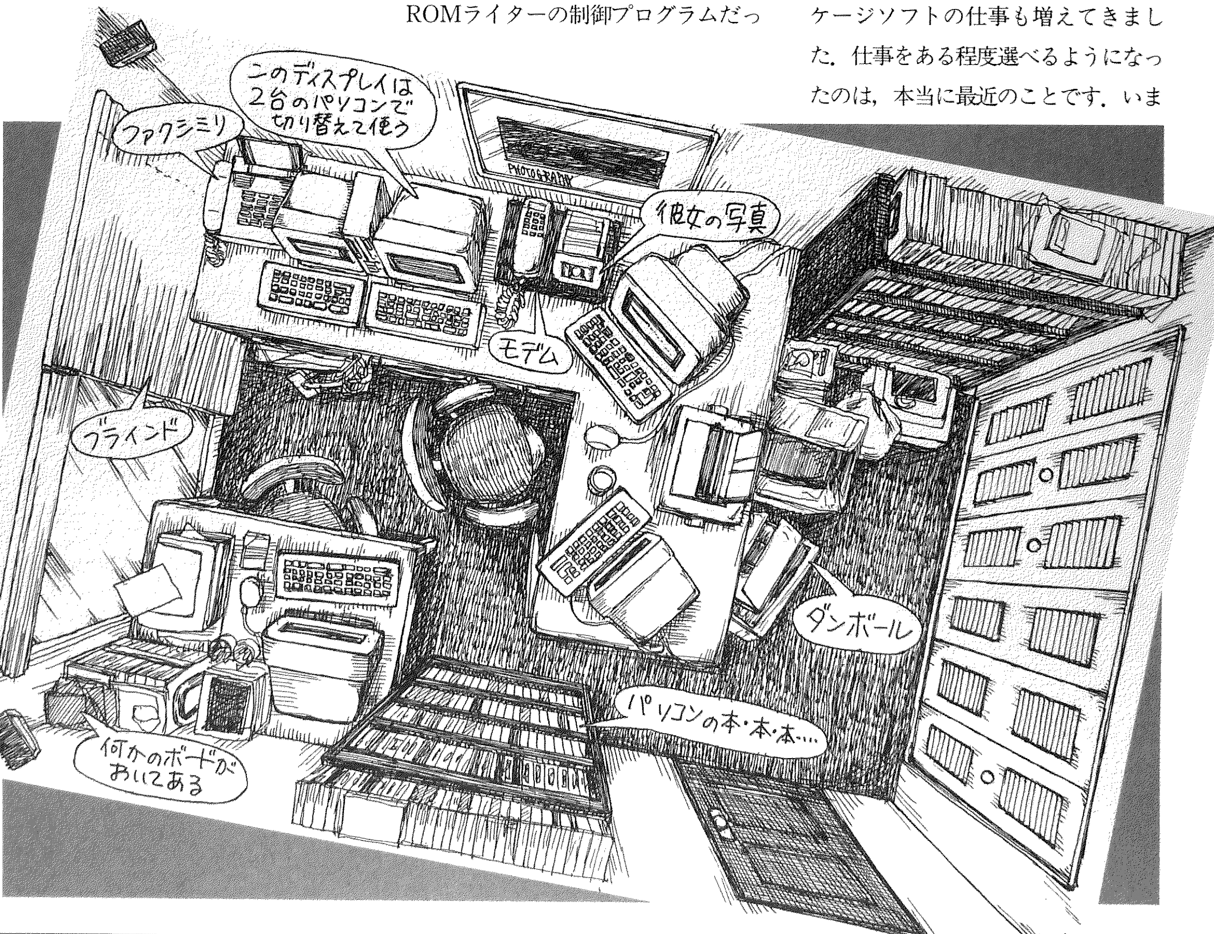
「UNIXマシンほしいけど，50万しかない．100万足りないね」↙

「それならうちの仕事やれば……」という調子で仕事が入ってしまったという．

そのときの仕事が，PC-98用EPROMライターの制御プログラムだった．この仕事で西川氏ははじめてC言語を仕事に使った．まだ，Lattice-Cが全盛期だった．

学生時代から数えると，もう8年もプログラムの仕事をやっていることになる．フリープログラマとして，就職とか組織とは直接関係のない世界で活動してきた．

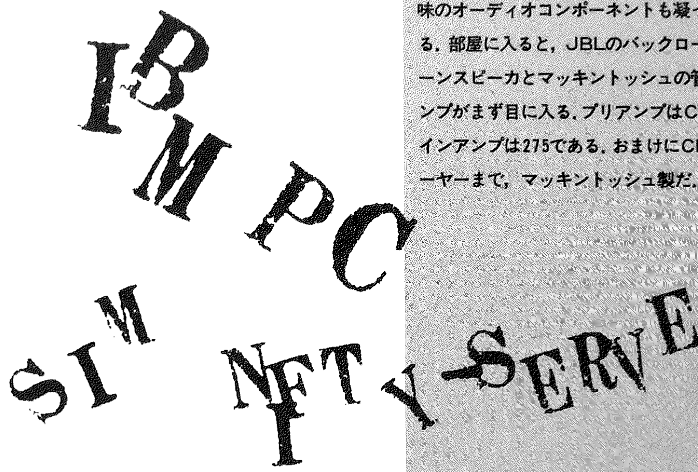
「つきあっている会社からくる仕事をやっていただけですから，ゲームからコントローラーの仕事まで何でもやりました．やはり，ハードウェアがらみのコントロールプログラムの仕事が多かったですね．ここ2年くらいは，パッケージソフトの仕事も増えてきました．仕事をある程度選べるようになったのは，本当に最近のことです．いま



までは，何でもやらなくては食べていかれませんでした」という，西川氏はいま転機を迎えている．

東京に拠点を移したことも，大きく状況を変える要因になったに違いない．質，量ともに増大していく仕事を前にして，新しい展開を考えなければならなくなった．

「いまはくる仕事を全部受けたのでは，時間がなくなってしまいます．そこで，会社組織にすることにして，この秋には仕事場を事務所に移します」．すでに，プログラミングの仕事の比率は半分くらいになり，企画や報告書の仕事が増えているという．新しいソフトウェア会社がもうすぐスタートすることになる．



Hobby

何事も徹底している性格なのだろう．趣味のオーディオコンポーネントも凝っている．部屋に入ると，JBLのバックロードホーンスピーカとマッキントッシュの管球アンプがまず目に入る．プリアンプはC22，メインアンプは275である．おまけにCDプレーヤーまで，マッキントッシュ製だ．

「これがなかったらいまの自分はまったく違った生活をしていたかもしれない」というものに，パソコン通信との出逢いがある．それは，米国をはじめとする巨大なソフトウェアの世界との出逢いだったといっていいだろう．パソコンネットワークは，さまざまな意味で新しいコミュニケーションの世界を切り開いた．ある人にとっては，内向した世界を形成するのにぴったりのメディアであったし，また，ある人にとっては翼を広げ飛び立つための舞台を提供してくれた．

1985年の，いわゆる通信自由化以降に開局したパソコン通信実験局のメンバーとなった西川氏は，まず，PC-VANでクロード・チアリがSIGオペをつとめるNEC98 SIG（チアリSIGと呼ばれる）を活動の場とした．当時のチアリSIGは，SIMプロジェクトで盛りあがっていた．SIMとは，98上にIBM PCの環境をシミュレートするプログラムで，田中氏の作品である．SIMを常駐させた上で，パッチを当てたPC用ソフトウェアが動作する．チアリSIGでは，PC-OUTLINE，PRO COMMなどPC用ソフトウェアのパッチ情報が次々と公開され，多くの人がPCソフトウェアにふれる機会を持った．

パソコン通信がなかったら違った人生になっていたかもしれない

西川氏は，通信デバイスに8250A（注2）を搭載したPC-98用の拡張RS-232Cを企画するなど，チアリSIG上でのプロジェクトに積極的に参加していった．それが，西川氏のコンピュータ環境に大きな変化をもたらすことになる．PC互換機と米国ソフトウェアへの関心が急速に強くなっていたのである．当時，米国で評判を呼んでいたコンパイラ，Turbo Cを知ったのもSIM環境上でのことであった．趣味の色彩が強かった米国ソフトウェアへの興味も，開発用ソフトウェアが登場するにおよんで，もはや趣味の段階を越えようとしていた．

また，ネットワークやそのオフラインミーティングで知り合った人たちとの交流は，西川氏の生活にも大きな影響を与えた．Nifty-Serve，日経MIXと交流の輪も広がっていった．

SIMで知った米国ソフトは，PC/XT互換機を入手することによって，

（注2）8250A
通信用のデバイスとしてIBM PC系のパソコンには8250Aが使われている．PC-9800系パソコンは8251であり，8250Aの機能に依存した通信プログラムはSIMを使っても，PC-98上で動作しない．

ますます魅力的に感じられた．西川氏の興味は，米国ソフトが動くパソコンに移ってゆき，AXやJ-3100を使うようになった．もちろん，これらのPC互換機では，米国PCのソフトウェアがそのまま動くからだ．米国のソフトウェア，それも最新のものを動作させることができるコンピュータ環境が西川氏にとって最優先課題となった．また，それらを快適に動作させることにも力を入れた．

たとえば，AXパソコンが登場すると，すぐにCFCというプログラムを作り，ネットワーク上に公開した．AXは，日本語モードでも多くの米国ソフトウェアが動作するが，米国のキャラクタ罫線などと日本のカタカナコードがぶつかるため，画面表示が見にくくなる．そこで，フォントを交換して日本語モードでも美しい画面表示を実現するのがCFCというプログラムの機能だ．

✚

開発用・企画書用・移植用など マシンは用途によって使いわける

仕事場にはパソコンがひしめいている．

IBM PS/55 5570S
IBM PS/55 5530T
シャープAX386
三洋MBC-18
東芝J-3100SS
エプソンPC-286UX

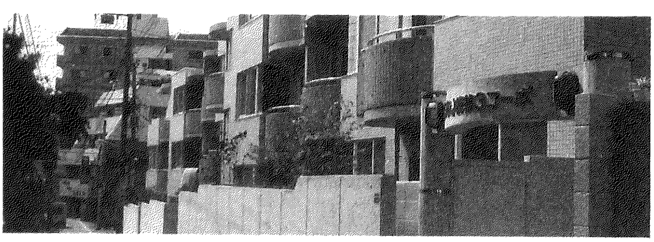
ひとりで使うにはあまりにも多くのマシンに思えるが，ちゃんと使い分けがされている．メインは5570Sで，ハードディスクには開発用のコンパイラやツール類，企画書や報告書を作成するためのワープロ（UP2）などが収められている．開発には，やはり5570SのOS/2環境がいいそうだ．このマシンで，ワープロを使うのは，17インチの大型ディスプレイがついているため表示が見やすいからだ．5530Tにも，同様の開発環境が収められているが，こちらはWindows 3.0がインストールされており，ほとんど英語モードで動かしている．通信を行うのもおもに5530Tだ．

AXは，日本語モードで米国のソフトウェアを動かしてみるのに適している．こうすれば日本語が通るかどうかなど，日本語版への移植にどのくらいの手直しが必要なのかをだいたい感じとることができる．

PC-286UXは，STARFAXを入れてFAXサーバーになっているし，東芝のダイナブックは「ベッドトップの通信用」なのだそうだ．

プログラムのノウハウは 結局個人に帰属するものなのだ

西川氏の開発ノウハウを聞くと，なるほどパソコン上のプログラミングというものは，ほんとうに個人に依存するものだな，と感じさせられる．資産として保存してあるルーチンを組み合わせることによって，プログラム開発の効率を高めている．これは，もちろん彼の長年の仕事によって培われた成果なのであるが，企業の開発部門でこれを行うのは難しい．なにがどこにありどう使うのか，それを多人数で複数のプロダクトが動いている中で適切に割り振ることは不可能に近いだろう．個人の能力によって効率化を押し進めることができる，これがパソコンプログラミングの世界なのだろう．



西川氏は，キーボード，ディスプレイなどのデバイス別のルーチンを数多く保存し，新しい仕事を受けたときは，できる限りそれらを組み合わせて，開発の効率化を図っている．新しい仕事の仕様を説明されているときに，彼の頭の中では「(とってある) ルーチンを

バックアップは自分で作成したファイルだけ バックアップツールは米国製がベスト

どのくらい使えるか」が計算されているのである．

それらのルーチンは，機種に依存した部分を省き，ソースコードの形で保存されている．オブジェクトファイルは，リンクすればすぐ使える反面，機種依存する部分も含まれてしまうから汎用性がない．西川氏は，汎用性のある部分だけにシェイプアップしたルーチンを保存しておいて，使う必要があるときは，ターゲットマシンに依存する部分をソースコードとして書き加えてプログラムを組み上げる．

そうした合理性は，開発環境全体にわたって感じられる．コンパイラにしてもそうだ．彼の開発マシンは，PC(またはPS/2)互換機だから，米国で発売されたコンパイラを入手すればすぐにでも使える．今年，MS-C 6.0とTurbo C++が出たときもそうだった．FAXオーダーで発売間もないソフトを購入しインストールして，その環境が気に入れば，たちまち実際の開発に使ってしまう．「日本語が必要な部分は，いままでの環境で作って後からリンクすればいい」のだ．コンピュータ環境に対する興味と仕事が食い違うことなく，並立していることも，彼の資質がパソコンという環境を得て生かされていることの証明だろう．

このような開発環境と方法は，企業内のプロダクトに応用するのは難しいだろう．個人と比べて，とてつもなく膨大な量のルーチンをモジュラー化して利用するシステムは，たいへんな労力を必要とするし，ターゲットマシン，使用する言語，コンパイラなどがある程度限られていなければ不可能だ．また，たとえ個人であっても，仕事の内容によってはルーチンの応用がきかない場合も多いだろう．↙

しかし，企業での開発環境においても，社員それぞれが使うハードディスク内のディレクトリやエディタをはじめとするツール類の構成などは個人に任せられていることが多い．一定の条件が揃えば試してみる価値のある方法である．

✚

ディレクトリは，図1のような区分で作っている．USERから下のディレクトリのファイルは，自分で書いた文書やソースファイルなので，1日に1回はバックアップをとるようにしている．サブディレクトリUSERは，その下にさらにサブディレクトリを作り，バックアップを容易にできるようにする．バックアップをとらなければならない部分は，だいたい5MBほどだという．

バックアップは，フロッピーディスクにとる．バックアップツールは，米国のソフトウェアにすぐれたものが多く，この分野では日本のソフトはかなわない．西川氏が使っているのはFAST-BACK PLUSというPCソフトウェアだ．

✚

西川氏の開発システムは，今後もパーソナルシステムの発展にともなって変化していくだろう．会社を設立してからの活躍が期待される．

図1：ディレクトリ構成

```
Y ---- ARC
     |- DEV
     |- DOS
     |- ETC------------- CONNECT
     |              |- MAXLINK
     |              |- MIFES
     |
     |- TEL
     |- TURBOC++ ------ SYS   (TURBOC++)
     |- MSC ----------- BINB   (MS-C 6.0)
     |              |- INCLUDE --------- MT --------- SYS
     |              |- LIB
     |- WINDOWS
     |- WIN3
     |- WINDEV
     |- WINDDK
     |
     |- Turbo Debuger
     |
     |- Tool
     |- USER ---------- DOC
                    |- CFC
                    |-  (仕事1)
                    |-  (仕事2)
                    |-  (仕事3)
                    |-  (仕事4)
                    |-  (仕事5)
```

〈ARC〉BBSなどのネットワークからダウンロードしたファイルを一時的に収めておく．
〈DEV〉デバイスドライバ類をまとめて収める．ただし，ルートディレクトリにないと機能しないものは除く．
〈DOS〉DOSに付属するコマンドファイル類を収める．デバイスドライバなどは別にまとめる．
〈ETC〉一般的なツール類．
〈TEL〉通信ソフトとログファイル．
〈TURBOC++〉Turbo C++コンパイラ．
〈MS-C〉MS-Cコンパイラ．
〈WINDOWS〉日本語MS-Windows 2.11
〈WIN3〉米国版MS-Windows 3.0
〈WINDEV〉MS-Windows開発用プログラム
〈WINDDK〉MS-Windows開発用
〈USER〉自分自身で作成したプログラムのソースファイル，文書ファイル，データファイルなどを収める．

プログラマの仕事場 | 5

Programmer's Working Space

ジィスクソフトウェア株式会社

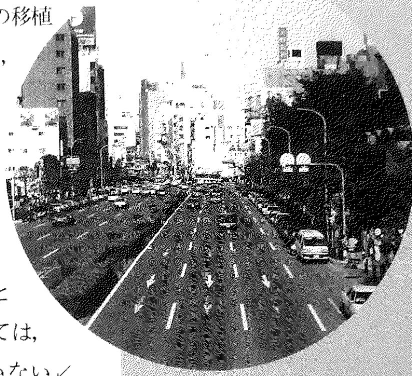
Windowsのパッケージソフト参入で一躍注目を浴びはじめた古参ソフトハウス

OS/2のプレゼンテーションマネージャ(PM)が，GUI(グラフィカルユーザーインターフェース)の世界へと扉を開いたのに続いて，MS-Windows 3.0が登場，GUIはますます現実的になりつつある．PM，Windowsといった新しい環境でのアプリケーション開発の舞台には，これまでとは違った役者が次々と名乗りを上げている．ジィスクソフトウェアもその中のひとつであり，米国Micrografx社などのソフトウェアの日本語化に取り組んでいる．

ジィスクソフトウェアという社名を一度でも聞いたことがあるという読者は，そう多くはないだろう．ジィスクソフトウェア株式会社は，本誌の特集で取りあげたほかのソフトウェア会社とは違って，おもにメインフレームのシステムプログラム開発を中心に活動してきた．現在では，その高い技術力をUNIXワークステーションやパーソナルコンピュータの分野でも発揮している．

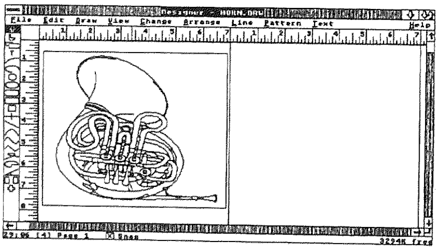
メインフレーム，ワークステーションなどで実績のあるジィスクソフトウェアが，なぜWindowsという未知のプラットフォームに進出したのか？その経過を営業統括部営業第２課課長，阿部文彦氏は次のように説明してくれた．

「メーカーからの依頼もありまして，海外でのWindows関係について，4，5年前から調査をはじめました．そのうえで，Windowsは伸びるだろうということでスタートしました．はじめは（米国のソフトハウスと）提携するという話はなかったんですが，最終的にはジィスクが（日本語版の移植を）やるという話になりまして，現在はMicrografx社などいくつかの会社と提携しています」．



Windowsアプリケーションを自社開発するのではなく，あえて米国ソフトの移植という手段をとったことについては，「Windowsがまだ普及していない状態で，Windowsのアプリを作るよりも，米国市場でシェアを広げているアプリケーションを移植したほうが早い．その中で技術も習得してアプリケーションを作っていけばいいんじゃないかということです」(阿部氏)．

米国アプリケーションの移植で Windowsの普及と技術の習得の一石二鳥



Micrografx Designer

Location

ジィスクソフトウェアの住所は渋谷区渋谷２丁目，JR渋谷駅から宮益坂を上って表参道へ向かう途中を少し脇道に入ったところだ．近くには青山学院大学，こどもの城，国連大学ビル（建築中）などがあるが，表参道，青山といえばやはりファッションの街．オフィスのあるビルの周辺もアパレルメーカーが多い．

About Company

ジィスクソフトウェア株式会社．1970年6月設立．浦山清治社長．

メインフレームのシステムプログラムからスタートした．コンピュータハードウェアの発展とともに，ミニコン，ワークステーション，パソコンへと守備範囲を広げている．システムプログラムの企画，研究，開発，製造，販売，コンサルティングがおもな仕事．

COLUMN

ハイパーテキスト

人間の思考は本を読むときのように一定の方向へばかり進むとは限らない．ひとつの情報から別の情報やイメージを連想したり，ときには音楽と結びつけたりすることもある．ものを書く側から見るときは，さまざまな情報をリンクさせ参照や注釈を自由に付けられる，書かれたものを読む側から見るときは，自分の発想のおもむくまま自由に読み進んでいける，それがハイパーテキストである．

はじめてこの言葉を使ったのは，米国のテッド・ネルスン（Theodor.H.Nelson）で，世界中のあらゆる文献資料をコンピュータで検索，読むことができるだけでなく，誰もが執筆できるようにするという構想のプロジェクト・ザナドゥを進める過程で提案された．

Windows 3.0で注目されたToolBookも日本語化が進行中

ジィスクソフトウェアが最も力を入れているのは，パッケージソフトの分野である．米国Micrografx社のWindowsアプリケーションについては，着々と日本語化（表1参照）が進んでいる．その第1弾がMicrografx社の主力製品のひとつ，Designerの日本語版だ．Designerは細密画用のイラストレーションソフトで，米国ではExcelやPageMakerと並び，人気の高いWindowsアプリケーションのひとつである．これらのWindowsアプリケーションについては，Windows3.0に対応することに照準を合わせていくことにしている．

表1:Micrografx社のWindowsアプリケーション（日本で発売または発売が予定されているもの）

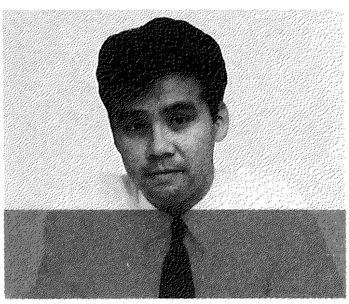
Micrografx Designer	細密画用イラストレーション
Micrografx CHARISMA	ExcelのDDEをサポートしたプレゼンテーショングラフィックックス，ドローイングソフト
Micrografx Clip Art	クリップアート集
Micrografx Xport	CGM,PICT,DXF,DRW,EPS,TIFF,GEMファイルのコンバート

ジィスクソフトウェアのWindows3.0への対応で，もうひとつ見逃せないのは，ToolBookの日本語化だ．ToolBookは米国Asymetrix社（注1）の開発したオブジェクト指向型のプログラム開発ツールである．米国ではWindows3.0のパッケージを購入すると，スペシャルボーナスとしてToolBookのランタイムバージョンと，ToolBookで作成したスケジュール管理プログラムDayBookがついてくるということで話題をまいた．

ジィスクソフトウェアでは，ToolBookに関するノウハウは1社で独占するのではなく，積極的に社外のプログラマにも広めようという方針だそうだ．近い将来には，

（注1）Asymetrix社
米国Microsoft社の創設者のひとりでもあるポール・アレン（Paul Allen）が設立した会社．

PROFILE



緒方秀俊

1962年5月，福岡で生まれる．大学までは福岡に．商学部の統計関係のゼミナールではじめてパソコンに触れる．卒業後，銀座に本社のあるソフトウェア会社に入社のため上京．その後ジィスクソフトウェアに．趣味はとくにない．時間があまりないのでコンサートを見に行く程度．家にはパソコンをおかない．



WindowsやOS/2のマニュアルと並んで，次のような本が資料として積まれていた．
「プログラミング言語C 第2版」(Kernighn&Ritchie著　石田晴久訳　共立出版)
「応用C言語」(三田典玄著　アスキー)
「Windows3 Companion」(Lori Lorenz and R.Michael O'Mara著 Microsoft Press)
「ToolBook Companion」(The Cobb Group:Joseph R.Pierce著 Microsoft Press)

ToolBookを使ったアプリケーションが続々と各社から登場する可能性もある.

✚

英語版のソフトを日本語で使えるように移植するという作業，実際にはどんなことをするのだろう？　「比較的移植は楽だと思いますね」と語るのは，Windowsアプリケーション移植作業を実際に担当するプロジェクトリーダーの緒方秀俊氏.

「プログラムの基本的なロジックとかデータの管理に関して，さわる部分はあまり多くありません．とくにWindowsアプリケーションの移植に関しては，目に見える部分がほとんどなんです．画面のメニューを日本語に直すとか，かな漢字変換での入力を受け付けるようにするとか．あとは漢字のデータをきちんと管理できるようにチェックを増やす．その程度です．とくにメニューまわりは，メニューやダイアログボックスの制御情報が1個の別のファイルにまとめてあります．そのファイルを日本語にしてしまうだけで，9割がたは問題なく終ります」「それより，アメリカで使っている用語を日本でわかりやすい用語に直すかということに一番時間を使います．ニュアンスを伝えるために，直訳したのとはまったく別の言葉で表すこともあります．たとえば，Designerの場合は，アメリカで使われている用語は日本でそのまま受け入れられない．Designerのターゲットがプロのイラストレーターやデザイナーであれば，デザイン用語にすべきだろうし，印刷を中心に考える人なら印刷用語にすべきだと思うんです．できるだけ広いユーザーに受け入れられるような用語を選定するっていうことはかなり大変です」.

移植での問題点はロジックより用語
プログラマには日本語のセンスも要求される

ジィスクソフトウェアでは，こういったメニュー用語の翻訳もすべてプログラマの仕事である．印刷に詳しい社員に聞いたり，本を読んで調べたり，ソフトウェアのパッケージを依頼したデザイン会社に質問したこともあるそうだ．プログラマに幅広い知識が要求される仕事の例である.

✚

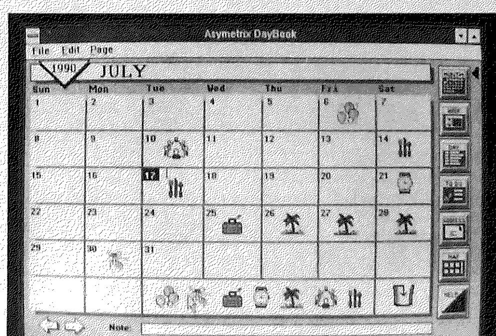
ジィスクソフトウェアではメーカーとの関わりが深いこともあって，開発中のソフトウェアのセキュリティには気を配っている.

「開発中のアプリケーションについては，鍵のかかるロッカーに入れることになっています．ハードディスクなどに入っているものについては，プログラマの管理にまかせていますが，基

COLUMN

ToolBook1.0

読者の中には「オブジェクト指向型の開発ツール」というと，何やら難解なプログラム言語を想像する向きもあるかもしれない．だが，ToolBookについては，開発ツールというよりも「カード型データベースのカードの連結の仕方を自分で自由に登録できるようなもの」をイメージしてもらうといいだろう．本格的なWindowsアプリケーションを開発するには，C言語やアセンブラの知識が必要だが，このToolBookを使えば一般のエンドユーザーでも工夫次第でアプリケーションを作ることができる．この種のソフトウェアは「ハイパーテキストソフト」と呼ばれ，マルチメディア時代のソフトウェアとして期待を集めている.



開発中のアプリは鍵のかかるロッカーへ
セキュリティ重視の開発環境

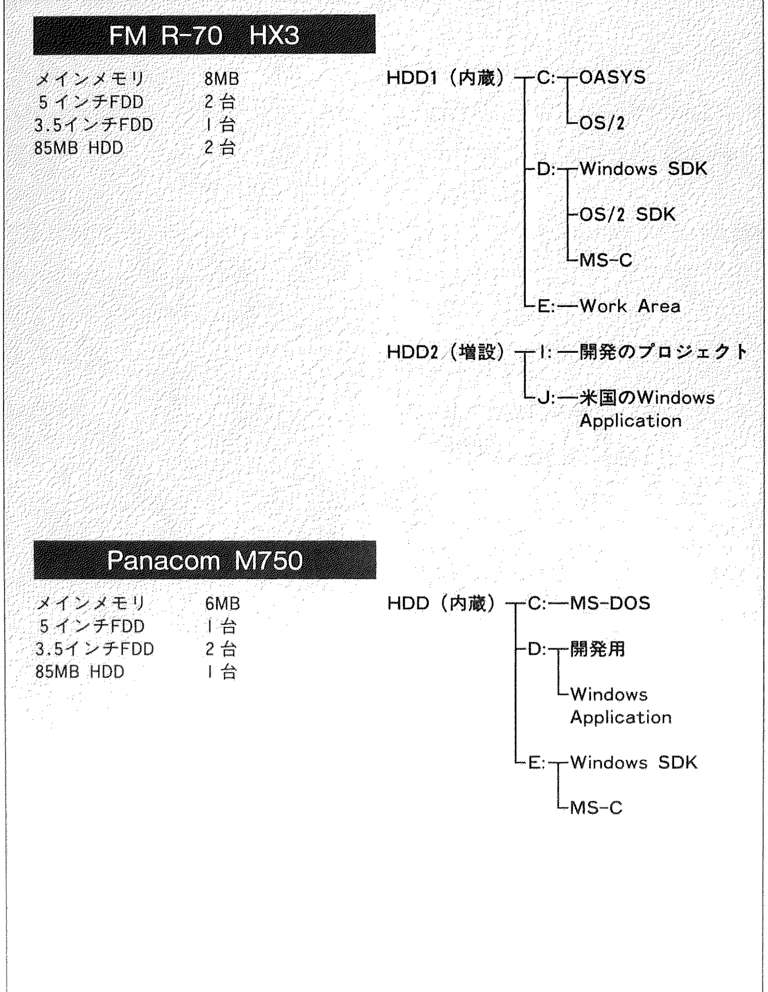
本的に開発している部屋には外部の者は入れないことになっていますので，それでセキュリティは守られています」（緒方氏）．

プログラムのバックアップについても，多いものでは毎日，あるいは週に1回，開発の区切りごとなどにそれぞれのプログラマが管理している.「結局バックアップがなくて泣くのはプログラマですからね」と緒方氏は語る．

次に緒方氏の開発環境を紹介しよう（図1参照）．開発マシンは，FMRが多数派である（Panacom M750は↙FMR-70 HX2と同じ）．目を引くのは，メモリ搭載量がが8MB, 6MBと多いことだ．このうち1MBのみをMS-DOSのメモリとし，残りはすべてRAMディスクとして使用している．この大量のRAMディスクには，コンパイラやWindowsのテンポラリファイルをおいて開発のスピードアップをはかっている．ソースプログラムは，ハードディスクにおいたままで開発するが，最初にマシンを起動したときに，開発中のソースプログラムをすべてRAMディスクにコピーして，ハードディスクが壊れた場合のバックアップとして利用している．バックアップについては，ほかにも2日に1度，必ずフロッピーディスクに保存することにしているそうだ．

エディタは，緒方氏も含めてほとんどのプログラマがREDを使っている．

図1：PMとWindowsの開発環境（緒方氏のシステムとハードディスクの内容）

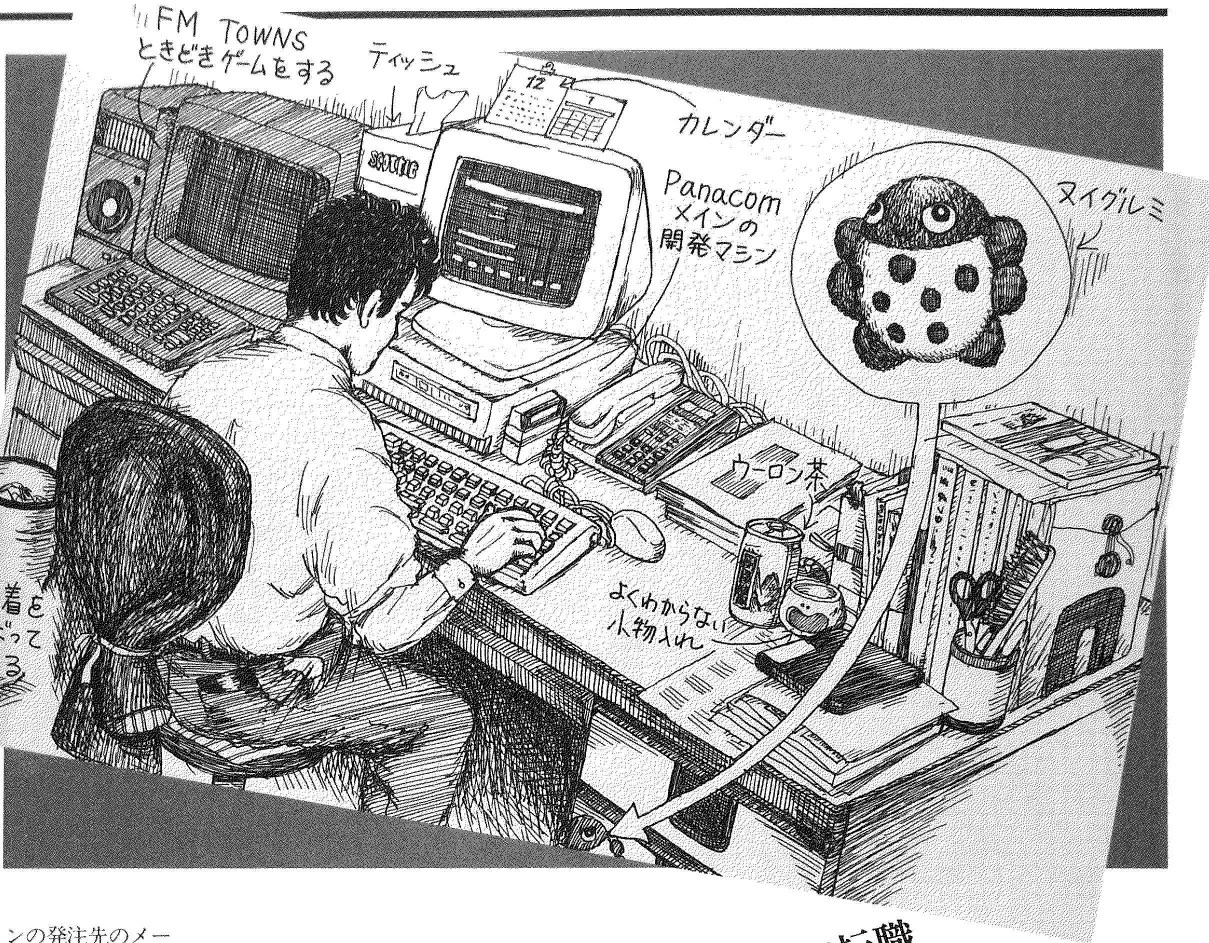


✚

いまやプロジェクトリーダーとして活躍する緒方氏だが，ジィスクソフトウェアではまだ3年少々の転職組だ．前の会社でもプログラマだったが，よりシステムに近いMS-DOSのアプリケーション作りたいと，デューダした．

「Cはジィスクソフトウェアに入ってからはじめてやりました．カーニハンとリッチーの本は読みましたけど，その後はプログラムを組みながら，わからないところはまわりの人に聞いて2週間程度で覚えましたね．前の会社でPL/1をやっていましたので，それほど抵抗はありませんでした」（緒方氏）．

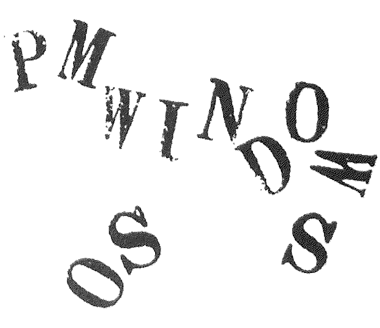
「ジィスクに入って，まずイメージ処理のアプリケーションをやりました．それと同じ時期にそのアプリケーショ



パソコンのシステムプログラマをめざして転職
PM, Windowsで自分の可能性を伸ばす

ンの発注先のメーカーにいきまして，そこでOS/2の勉強ですとか，移植の手伝いですとか，テストプログラムの改造，移植などをやりました．イメージエディタの手が離れたころから社内に戻って，引続きOS/2の関係の仕事をしばらくやりましたが，Designerの移植が決ってWindowsに移りました．PMをやっていましたから，Windowsには違和感なく移れましたね」．

「これからWindowsのアプリケーションは伸びていくと思いますし，日本ではまだWindowsのプログラムを作る人が少ないと思うんです．何百人とか何千人ぐらいだと思うんですけれど，その中で当面は，Windowsを自分のものにしていきたいですね．Windowsを結構やってきたつもりではいるんですけれど，まだまだ知らないことって多いと思いますし．Windowsといえばジィスクソフトウェアという評価を受けたいですし，Windowsアプリケーションの世界でプログラマといえば緒方だといわれるようになりたいんです」と語る緒方氏の口調には，着実にスキルアップしているという技術者の自信が感じられる．



プログラマの仕事場 | 6

Programmer's Working Space

株式会社演算星組

道具としてのソフトウェアと情報としてのソフトウェアを考えながらマッキントッシュ中心の開発を進める

毛筆シミュレーションソフト『Mac書道』，ハイパーカードのスタックウェア『サイバースペースデッキ』などで知られた演算星組は，マッキントッシュによる開発をおもに行っている．マッキントッシュの開発環境は，この特集ではただひとつだ．当然，DOSマシン上での開発環境とは大きく異なるし，開発者のソフトウェアに対する考え方自体に特色があるように思える．

現在，演算星組が開発しているのは，おもにマッキントッシュのアプリケーションだ．それはさらに，マックのCやPASCALを使って書くプログラムと，いまやマッキントッシュの標準的なソフトウェア環境となったハイパーカードのスタックウェア（注1）という2種類にわけることができる．

「マックのことなんかぜんぜん知らん」という読者のために，ちょっとMS-DOSの世界とくらべてみよう．CやPASCALなどの言語でプログラム↙

もうひとつのハイパーカードとは，Windowsにのった開発・実行環境であるGuideやToolBookにあたる（いずれもハイパーテキスト理論（注2）を基礎にしたソフトウェアである）．ハイパーカードなどのソフトウェアを使えば，それぞれの作法に従ってアプリケーションを開発することができる．データをのせる器を作るための部品や，データを関連づけるツールがそろっている環↙



一般的なプログラミング言語とハイパーカードによる2種類のアプリケーション

を組み，おもにシステムルーチンを使って作るのが，伝統的かつ一般的なマッキントッシュのプログラムだ．DOSでいったら，MS-WindowsのAPI(アプリケーションプログラムインターフェース）を使って書くプログラムにたとえられる．

境と考えればいいだろう．

位置的には，図1のようになる．MS-DOS上ですぐに実行できるプログラムと，MS-DOS上のBASICやdBASEのようにインタプリタ的な環境を必要とするプログラムの差にたとえられる．

（注1）ハイパーテキストなどのために作られたアプリケーションは，特有の呼び名をもつ．ハイパーカードではスタックウェアであり，Guideではガイドライン，ToolBookではXXXBook（たとえばDayBook）である．

（注2）ハイパーテキスト理論（59ページ参照）

図1：一般的なアプリケーションとハイパーカードなどの違い

●一般的なアプリケーション

- アプリケーション
- システムソフトウェア（MacOS，Toolbox／MS-DOS，Windowsなど）
- ハードウェア

●ハイパーカードなど

- アプリケーション（スタックウェア，ガイドライン，XXXBookなど）
- 開発・実行環境（ハイパーカード，Guide，ToolBookなど）
- システムソフトウェア（MacOS，Toolbox／MS-DOS，Windowsなど）
- ハードウェア

Location

千駄ヶ谷の国立能楽堂の近く．どちらかというと住宅街の一角にあるマンションの一室に演算星組の事務所はある．薄暗くライティングした室内は，やはり普通のソフトハウスとは少し違う雰囲気をかもしだしている．

About Company

おもにマッキントッシュ上のソフトウェア開発を手がけるソフトハウス．創立は1985年．マッキントッシュという日米共通のマーケットを標的とする「Mac書道」などのソフトウェアは，米国のマックユーザーにも支持を受けている．

しかし，マッキントッシュ専門のソフトハウスではなく，PC-98上の「風」はかな漢字入力ソフトとしてユニークな存在である．最近では，ハイパーカードのスタックウェアをMS-DOS，WindowsのGuideに移植するような仕事もあるという．6人の社員がそろって開発に取り組んでいる．

Products

電脳絵巻シリーズ：「天の巻」「地の巻」「器の巻」

電脳文具シリーズ：「Mac書道」「PC書道」「電脳手帳」

ハイパーカードスタックウェア：「サイバースペースデッキ」

超多段シフト日本語入力ソフト：「風」

ハイパーカードのような環境では，情報を表示したり関連づけたりするためのルーチンをプログラムする必要はなく，いかに情報自体を組織していくかが仕事の中心となる．MS-DOSマシンのプログラミングにおいても，最近はそういった傾向になってきている．Windowsの普及を例にとるまでもなく，パーソナルコンピュータの十数年にわたる歴史の蓄積は，マシンの機能を深く知ったうえで，ひとつひとつのルーチンから書かなければならないようなプログラミングを減らしてきている．さまざまなライブラリがゆきわたり，オブジェクト指向プログラムが語られる中で，たしかに流れが変わりつつある．従来のようにプログラムのソースだけでなく，ソフトウェア自体が情報をもつようになったことで，情報のソースが問われる時代になろうとしているのだ．

かつて，画家自らが顔料を探し，絵の具を練らなければ絵を描けなかった時代があった．やがて，絵の具メーカーと画家は分業することになった．ソフトウェアの世界でも，システムや文具にあたるソフトウェアを作る人と，その上に盛る情報をまとめあげる人は別の道

マッキントッシュの環境は自分たちのやりたいことを実現してくれる

COLUMN

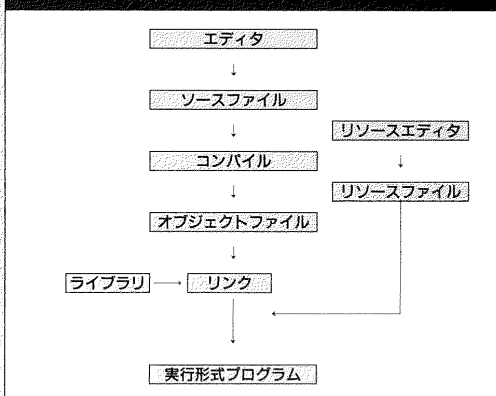
マッキントッシュのプログラミング環境

ほとんどの処理系が使えるようになった

マイコンキットにはじまるパーソナルコンピュータの歴史の中で，マッキントッシュは特殊な位置を確保しながら普及してきた．誰にでも使えるコンピュータをテーマに登場したマッキントッシュは，グラフィックユーザーインターフェースやマルチウィンドウによってエポックを作りだしたが，当初はお絵描きソフトとワープロのプレーヤーでしかなかったというみかたもできる．なにしろ，マッキントッシュ自体に開発環境はなく，上位機種であるLisa (Macintosh XL)を開発マシンとして位置づけていたのだから．

これは，プログラマブルであるのが当然であったパーソナルコンピュータの流れからはずれたものであった．もちろん，ユーザーからの反発も強かった．そうした状況が急速に変化したのは，Macintosh Plusの発売以降のことだった．アップル社あるいはサードパーティよって言語処理系の移植が次々と行われ，同時に，開発用のツールやユーティリティも充実していったのだった．それによって，通常使われるほとんどの言語がマッキントッシュで使えるようになった．マックの標準的な位置を占めるPascalをはじめ，C，BASIC，FORTRAN，COBOL，Modula-2，Prolog，LISP，LOGO，Smalltalk-80など，おもだった言語は使える状況になった．

図A：マッキントッシュのプログラム開発



そんな中で，マッキントッシュの標準開発環境といえるものがMPW (Macintosh Programmer's Workshop)だ．UNIXライクなシェルをもつ，マルチスクリーンの開発環境と68000系のマクロアセンブラを備えたMPW本体と，MPW Pascal，MPW C，Pascalのオブジェクトライブラリである MacApp などの各ソフトウェアセットから構成

を歩むようになるだろう．拡大していくマーケットでは，時計の針が進むのとともに分業化は不可避になっていく．

ワープロや表計算，ペイントソフトが一種の文具であるとすれば，これらはユーザーが文書を，データを，入力することによってはじめて「意味」をもつ情報を提供することができる．紙，筆，ペンなどが作られ，それによって書かれたものが，印刷技術に支えられて出版という仕事を可能にしたように，パソコンの文具（道具）を開発することが中心であった時代から，パソコンの上で読む「本」が製作される時代へ進もうとしている．

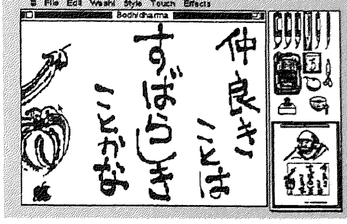
そういった意味で，マッキントッシュの環境は先進的であるとみることができる．文字，音，そして図像を扱うために，マッキントッシュは一般のMS-DOSマシン以上のものを用意している．演算星組がマッキントッシュにこだわるの理由もそこにある．「自分たちがやりたいことを可能にしてくれるからマッキントッシュを使う」のだ．

✚

道具としてのソフトウェアと情報としてのソフトウェア，演算星組はその両方とも開発している．現在開発して

Mac書道

筆と墨の織りなす枯淡のタッチをマック上でシミュレートするペイントソフトの一種．和紙と筆の組み合わせによる微妙なニュアンスの表現を実現したソフトウェアは，ほかに類を見ない．書道という名のとおり，筆字はもちろん水墨画のような絵を描くこともできる．米国での製品名は「MacCalligraphy」．



されている．マッキントッシュの開発ソフトウェアや技術資料は，APDA（Apple Programmer's & Developer's Association）団体から入手することができるようになっている．日本からは，国際会員として米国のAPDAに加入することはできるが，日本の国内組織はない．

MPWは，標準的で公式な開発環境とされ，プロのツールというイメージが強い．そこで一般ユーザーがプログラミングをするときは，サードパーティから発売されていて，日本でも秋葉原のショップなどで入手できるTHINK Cなどの処理系を使うことになる．

独自のシステム環境

マッキントッシュの開発では，MS-DOS上のそれのように，プログラマがDOS，BIOSなどの機能を必要に応じて自由に使うことはできない．システムソフトウェアに用意されたルーチンを利用してプログラムすることが基本だ．

マッキントッシュは，システムソフトウェアとしてマッキントッシュ・オペレーティングシステム（MacOS）とユーザーインターフェース・ツールボックス（Toolbox）をもっている．これらは，アプリケーションとハードウェアを仲介するシステムルーチン群である．メモリ管理，ディスクファイル管理などを行う，よりハードウェアに近い位置にあるものがMacOSであり，その上位に位置するのがToolboxである．Toolboxは，グラフィック描画ルーチンやリソース管理，ウィンドウ管理などを行うルーチンから構成されている．

マッキントッシュのシステムソフトウェアの位置は，図Bに示すようになっている．これらのシステムソフトウェアの大部分がROMに収められている．ここで特徴的なことは，ROMのソフトウェアに加えて，起動時にシステムディスクからRAM上に拡張部分をロードすることだといえる．マックシステムのバージョンアップとは，ディスク上のシステムファイルのアップデートにほかならない．

プログラムからシステムルーチンを呼び出すときは，実アドレスを直接アクセスする必要はなく，トラップワードと呼ばれる命令で行われるから，ROM，RAMの区別に注意にする必要はない．

図B：マッキントッシュのシステム構成

アプリケーション
↓↑
ユーザーインターフェース・ツールボックス
↓↑
オペレーティングシステム
↓↑
ハードウェア

いる2種類のソフトウェアは，その典型を示しているといえる．ひとつは，「サイバースキャン」であり，もうひとつは「NihonngoWare」だ．

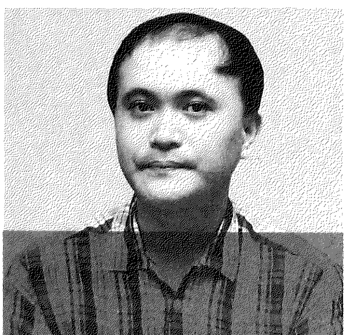
サイバースキャンは，カラーイメージスキャナから取り込んだ画像を処理するソフトウェアだ．サイバースキャンは，MPW（コラム参照）という開発環境でプログラムされている．MPWは，いくつかの開発用ソフトウェアを組合わせたセットで構成される．プログラマ天野賀章氏が使っているセットは，MPW Ver. 3.1，MPW Pascal Ver. 3.1，MacAPP Ver. 2.0という組み合わせだ．Cのときは，別の統合環境をもったTHINK Cを使う．ちなみにハードウェアは，本体Macintosh II（メモリ8MB），内蔵40MBのハードディスク，外付け60MBのハードディスクという構成だ．

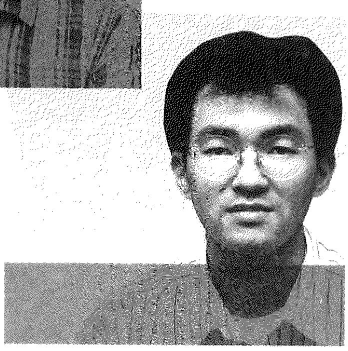
NihongoWareは，米国で発売する日本語教育用のスタックウェアで，2,300枚のカードと7,000にもおよぶ音声ファイルによって構成されている．データは180MBにもおよぶため，メディアにはCD-ROMが使われる．このスタックウェアは，演算星組が他社から受注したものであり自社製品ではない．日本での生活をシミュレートする感覚で，メニューに従って日本語の練習を行う．たとえば，交通の項目の新幹線の切符を買う場面では，窓口の絵がでて画面の日本人がこちらに話しかけるようになる．日本語はちゃんと音声で出力される．必要に応じて和英対訳をみたり，数の単位を参照することができる．

道具を作るソフトハウスから情報を組織するソフトハウスへ

このスタックウェアの原稿は，発注した会社から文字原稿として入ってきた．それをハイパーカードの環境の中でどういったユーザーインターフェースを作り，情報をどう関連づけるかといった作業を行うことになる．「そこがいちばん難しく，またおもしろいところでもあるんです」と，担当の岡野泉氏は話す．「マッキントッシュは，画面のイン

PROFILE





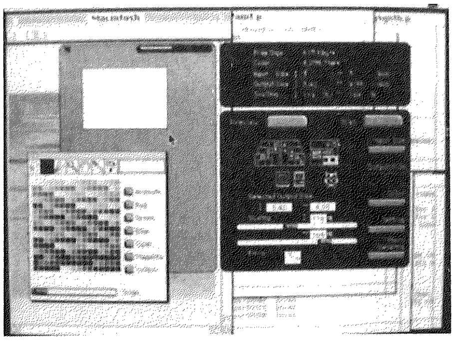
岡野　泉

出版，科学万博のパビリオンの企画などの仕事をへて演算星組へ．情報自体を扱うコンピュータ上の出版を行っていきたいという．

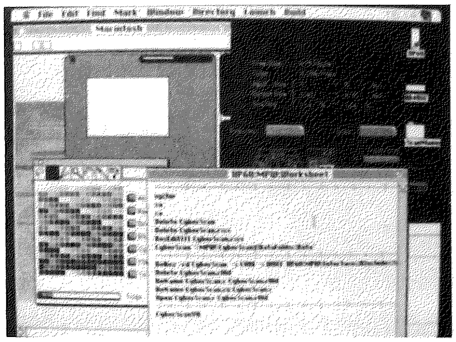
天野賀章

オフィスコンピュータのOSを開発する仕事から転職して演算星組へ．ユーザーインターフェースを大切にするマッキントッシュでの開発の仕事に満足しているようすだ．

ターフェースを作ることに力を入れやすい環境になっています．従来のMS-DOSのように1から自分で作らなければならないということがありません」．



開発中のサイバースキャン

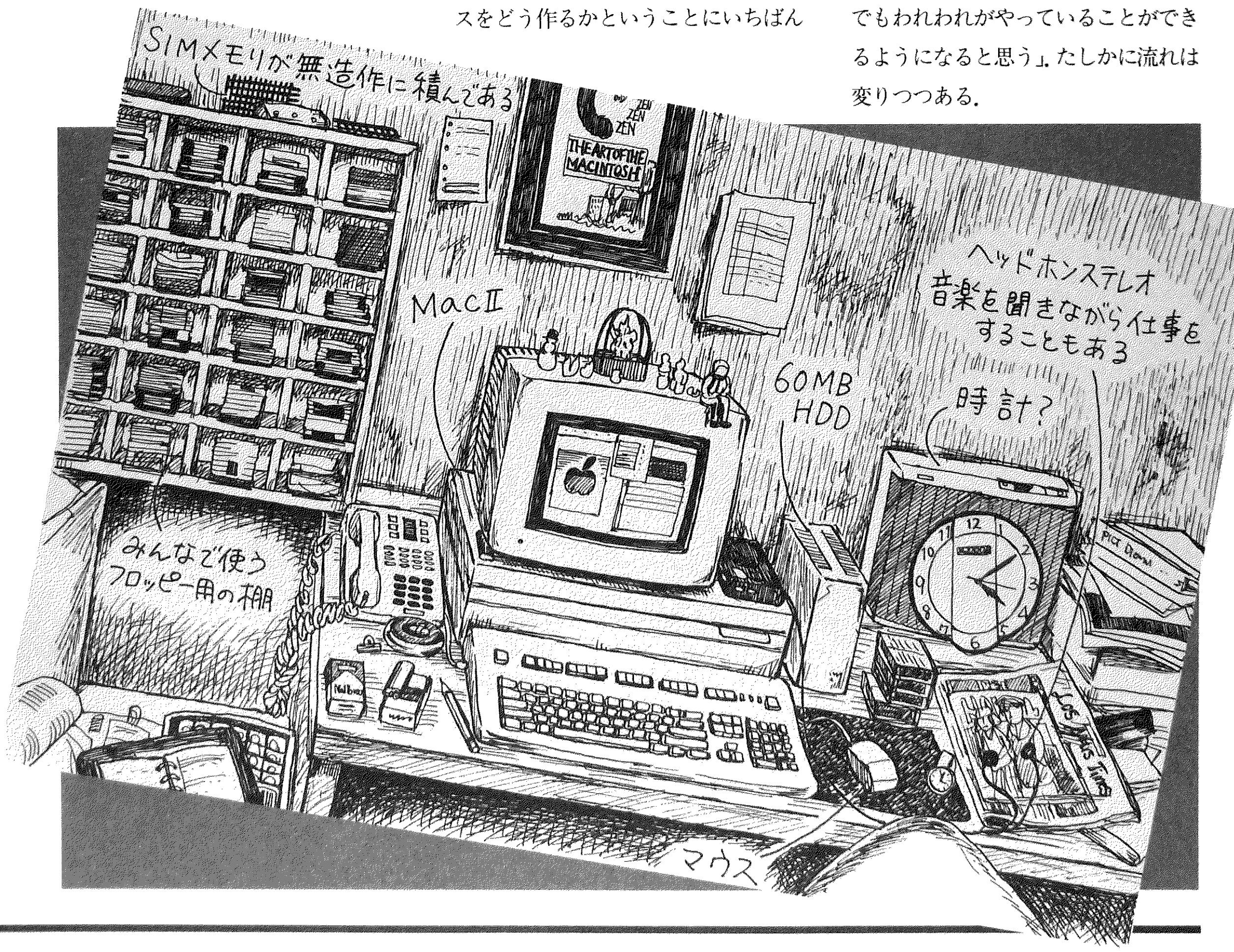


サイバースキャンにオーバーラップしたMPW

岡野氏は，Nihon-goWareの仕事をマルチメディアへのひとつの試みと位置づけている．「現在のマルチメディアは，ゲームなどのエンターテイメント中心になっていますが，その次にくるものとしてまず教育が考えられます．その分野でマルチメディアを生かしていこうというつもりで作りました」という．

サイバースキャンの開発を担当している天野氏も，「たしかにプログラミング上の制約の中でプログラムを組んだり，裏技を使ってみたりというおもしろさもありますが，直接画面にでて，ユーザーの目にふれるインターフェースをどう作るかということにいちばん頭を使います」という．

「うちの会社の開発は，文字や音，画像などいろいろな要素が入ってきます．マックでないとこうした情報を扱うことができなかったため，ここ4，5年はマックでやっていました．しかし，これからはMS-DOS系のマシンでもわれわれがやっていることができるようになると思う」．たしかに流れは変りつつある．



プログラマの仕事場 7

Programmer's Working Space

メガソフト株式会社

ユニークな開発支援ツールを生みだす開発環境はシンプル・イズ・ザ・ベスト

メガソフトは，軽快な操作性と強力なマクロ機能で知られるフルスクリーンエディタ「MIFES」や，RS-232Cを利用したリモートディスクシステム「MAXLINK」，ユニークな80386用メモリマネージャ「MEMORY-PRO386」の開発元である．
常に，プログラマの道具となるべき開発支援ツールを提供していきたいと語る，メガソフト開発部長高井護氏にお話をうかがった．

もともと高井護氏は，FA(Factory Automation）の会社でROMの組み込みシステムの開発をしていたが，当時売り出されたばかりのPC-8001を見て，これでなにかひとつやりたいと，オリジナル言語のコンパイラを開発し，パソコン雑誌に投稿した．このときの反響から，これはうまくすると商売になるのではないかと，前坂昇氏(現メガソフト社長）と共に作ったのがソフトウェアハウスメガソフトである．

パソコン雑誌への投稿がきっかけ これは商売になるとはじめたのがメガソフト

メガソフトがスタートした時点では，CP/M全盛だったことから，CP/M-80のコンパイラなど，開発者向けのツールを販売していた．

OSがMS-DOSに変わってからも，RAMディスクなど，アプリケーションではないちょっと変わったソフトを開発してきている．

MIFESを開発したときも，エディタなどというソフトは，プログラマだけが使うものだと思っていたため，現在のように多くの人が文書作成に使うようになるなどとは考えもしなかったと話してくれた．

MIFESが，今のようにワープロなのかエディタなのかわからない，速くて文字が入ればいいというような使い方をされることに関して，開発者としては若干の寂しさを感じるようだ．やはり，プログラマのためのエディタ，プログラミングのためのエディタを作りたいという気持ちはあるという．

メガソフトの基本は，プログラミング環境をよくしていくための道具作りにあるのである．

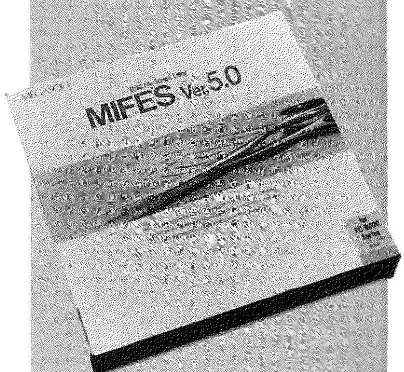
About Company

メガソフト株式会社

前坂昇社長．少数精鋭主義は，確固たる考えがあってのことだという．人数を増やして会社の規模を大きくするより，1人あたりの売上を増やしていく方針だ．同社の製品は，プログラマにおなじみのものばかりだが，最近は通信関係の製品も充実してきている．

Products

MIFES
MAXLINK
STARFAX
MEMORY-PRO386



COLUMN

OPTASM Ver.1.5 脅威の超高速アセンブラ

OPTASMは，MASM Ver.5.0の4倍のスピードをもつといわれる超高速アセンブラである．しかも，これだけのスピードを実現しながら，なお前方参照などではMASMよりも優れた最適化を行うというのだから，まさに脅威としかいいようがない．

8086の無条件ジャンプには，ショートジャンプとロングジャンプの2種類の命令があるが，OPTASMではこれらを記述するときにいちいち指定をしなくとも，ターゲットとの距離に応じて自動的に最適な命令を出力してくれる．

また，8086の条件ジャンプ命令はショートジャンプのみなので，その範囲外へジャンプする際には特別な記述が必要だったが，OPTASMでは特別な記述なしに最適なコードが生成される．

これらの機能のため，OPTASMを使用するプログラマは，ジャンプに関してほとんどなにも気にする必要がなくなる．また，OPTASMはMAKE機能を内蔵しているので，MAKEファイルを参照して連続アセンブルを行うことが可能である．

OPTASMの欠点としては，80386に対応していないことがあげられる．

開発マシンはスタンドアローン パラレルインターフェースでMOディスクを共有

ソフト開発に使用するメインマシンは，NEC PC-9801RLである．ほかには，PC-9801RAを補助的に使い，ターゲットマシンが必要な場合にはこれも用意する．だいたいプログラマ1人が，4～5台のマシンを使用している．

開発に使用するPC-9801には，すべてハードディスクが接続されているが，その容量はプログラマが開発している製品によってさまざまである．

LANなどの本格的な設備はないものの，パラレルインターフェースを用いて，プログラマ全員が1台の光磁気(MO)ディスクドライブを共有している．この光磁気ディスクドライブは，おもに開発中のソフトのソースのバックアップに使われている．この簡単なバックアップシステムをコントロールするソフトには，自社開発したものを使っている．

現在のメガソフトでは，プログラマもそれほど多いわけではないので，開

COLUMN

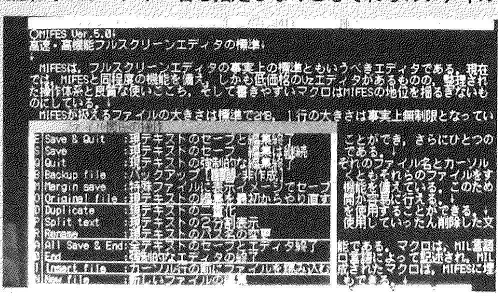
MIFES Ver.5.0 高速・高機能フルスクリーンエディタの標準

MIFESは，フルスクリーンエディタの事実上の標準ともいうべきエディタである．現在では，MIFESと同程度の機能を備え，しかも低価格のVzエディタがあるものの，整理された操作体系と良質な使いごこち，そして書きやすいマクロはMIFESの地位を揺るぎないものにしている．

MIFESが扱えるファイルの大きさは標準で2MB，1行の大きさは事実上無制限となっている．同時に扱えるファイル数は，15個である．

また，ふたつのファイルを同時に画面に表示して編集することができ，さらにひとつのファイルの別々の部分を同時に表示して編集することが可能である．

MIFESは，編集の終了時にオープンしていたファイルそれぞれのファイル名とカーソルの行位置を記憶し，次回の編集時にはファイル名を指定しなくともそれらのファイルをすべてオープンしカーソルを移動させるという，プロファイル機能を備えている．このため，複数のソースファイルを用いて編集している際の中断・再開が容易に行える．



検索・置換においては，日本語に完全に対応した正規表現を使用することができる．

また，各種の削除バッファを備えており，アンドゥ機能を使用していったん削除した文字列を画面に復帰させることが可能である．

MIFESのもっとも大きな特徴といえるのが，そのマクロ機能である．マクロは，MIL言語と呼ばれるBASIC風の文法とC言語風の演算子を備えたマクロ言語によって記述され，MILCというコンパイラによってコンパイルされる．こうして作成されたマクロは，MIFESに埋め込むことも外部マクロコマンドとしてファイルにすることもできる．

MIL言語には，メモリやI/Oを直接アクセスしたり，BIOSやデバイスドライバを呼び出す関数が備えられているため，かなり高度な機能が実現可能である．

また，MIFESでは各種のコンパイラが出すエラーメッセージに対応したタグジャンプが可能なため，強力なマクロと組み合せることでTurboCやQuickCのような統合環境を構築することも容易である．

発マシンはスタンドアローンで十分だということである．また，プリンタからソースリストを打ち出すようなことはほとんどないということなので，プリンタを共有する必要もとくにないようだ．

ターゲットマシンがある場合には，PC-9801RL上でコーディング，コンパイルを行った後に，MAXLINKを用いてバイナリファイルをターゲットマシンに送り込んで動作確認をしている．↙

✚

シンプルな環境なら事故も起きないし混乱も生じない

開発に使用する言語は，アセンブラとC言語である．ハードウェアに直接かかわるようなソフトを開発することが多いため，アセンブラの使用頻度はかなり高いようだ．

C言語のコンパイラは，TurboCかMS-Cを使っている．どちらを使うかは，プロジェクトの内容やプログラマの好みによって決められる．

アセンブラは，おもにOPTASMを使っているが，80386上のソフトを開発するときには，現状でOPTASMが80386に対応

COLUMN

MAXLINK　ほかのパソコンのドライブを外部ドライブに

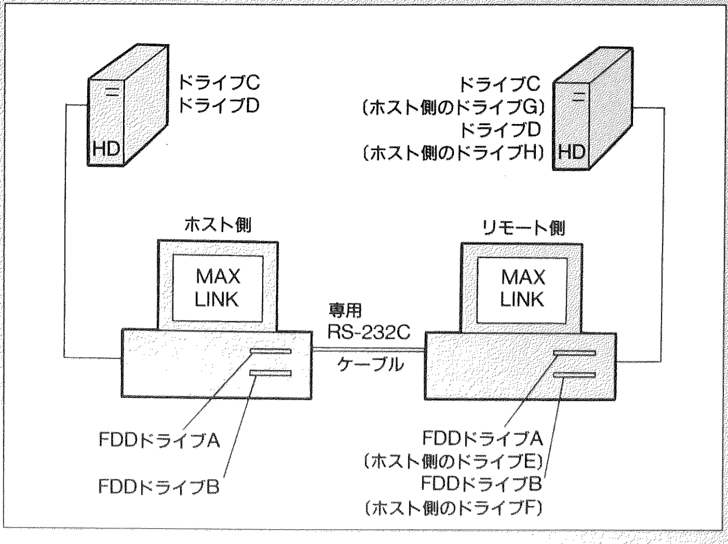
MAXLINKは，2台のパソコンのRS-232Cインターフェースを付属のケーブルで接続することで，一方のコンピュータのドライブをもう一方のコンピュータの外部ドライブとして扱えるようにするソフトである．

MAXLINKを用いることで，PC-9801とJ-3100など，異機種間でのファイル転送が簡単に実現できるようになる．

実際にMAXLINKを使用するには，まず付属のケーブルでホストになるコンピュータとリモート側のコンピュータのRS-232Cを接続し，次にホスト側のコンピュータにMAXLINK.DRVをデバイスドライバとして登録する．こうすることで，リモート側のドライブが，ホスト側に直接接続されているドライブの後ろにアサインされる．最後に，リモート側でMAXLINK.EXEというプログラムを起動すれば，リモート側のドライブはホスト側のコンピュータの外部ドライブとして使えるようになる．

MAXLINKの特徴は，その転送スピードにある．なんと，最低40000bpsから最高120000bpsという超高速転送が実現されているのである．通常，PC-9801の通信ソフトでは9600bpsが最高であるから，MAXLINKはその4～12倍の転送速度を誇っていることになる．また，MAXLINKはデータ転送の際にCRCによるチェックを行っているため，データが化けるというようなことはない．

ほかにも，MAXLINKのパッケージに入っているMAX.EXEというソフトをホスト側・リモート側の両方で起動することで，簡単にファイル転送をすることも可能である．



していないため，MASMを使用するということだ．OPTASMのいい点は，アセンブルスピードが速いところだ．

コーディングに使うエディタは，もちろんMIFESである．

MAKEは，TurboC付属のMAKEか，MS-C付属のものを使用し，MAKEファイルは，なるべくシンプルにして，凝ったことはしないようにしているということだ．これは，ただでさえソースが複雑なのにMAKEファイルまで複雑にしてしまっては，書いた本人は理解できてもほかの人間にはまったく理解できないものになってしまうからである．

ほかには，IBM PC版Norton Utilities Advanced Edition Ver. 4.5の中から，DOSジェネリックなものを取り出して使用している．

NORTON UTILITIES

また，コンパイラに付属している標準ライブラリはなるべく使わず，自社開発したライブラリを使用するようにしているが，これはあくまでもメンテナンスの問題で，標準ライブラリのできのよしあしとは無関係ということである．

現状では巨大なソフトウェアは作っていないので，複数のプログラマにプログラミングを割り当てるときには，モジュール単位ではなく，コマンド単位で割り当てるようにしている．つまり，ひとりのプログラマが完全に独立したプログラムを作成することになる．

ひとりのプログラマが一部のモジュールだけを作成するようにすると，プログラマがいったい自分はなにをやっているのかという意識がなくなるし，ソースの管理がたいへんになるのでよくないからである．

ソースのバックアップは，差分をとるような小細工なしで，バージョンごとに全部まとめてとるようにしている．RCSのようなものはいっさい使わない．そのほうが事故が少ないし，光磁気ディスクドライブがあるので，ソ

COLUMN

Norton Utilities Advanced Edition Ver.4.5 IBM PCユーザーの常備ユーティリティ

Norton Utilitiesは，IBM PCユーザーならば，あの腕組をしたMr.Nortonの姿と共に知らぬものはいないというほど有名なユーティリティソフトである．

Norton Utilitiesは，26本の小さなユーティリティソフトをひとつのパッケージに収めたものである．それらの多くはハードディスクを活用するためのツールである．

Norton Utilitiesの中でもっとも有名なのは，削除したファイルを復活させるNU (NortonUtility) というソフトだろう．NUは，エコロジーなどのツールでは復活できない削除ファイルを，すべては無理でもできる限り復活させるという，貴重なファイルをまちがって削除してしまったユーザーにとっては最後の頼みの綱ともいうべきソフトだ．

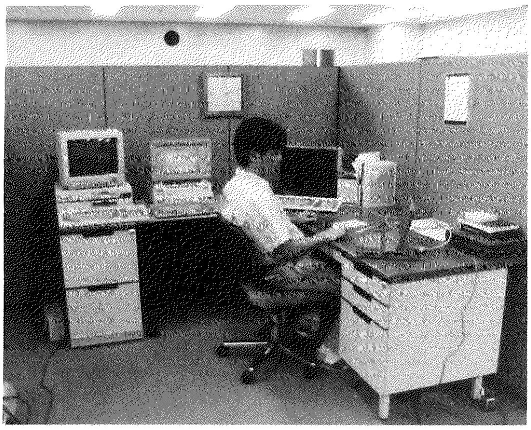


NUを使えば，消去エリアを削除したファイルに含まれていた文字列をキーワードにサーチし，必要なクラスタを集めてひとつのファイルにまとめることができる．

ほかにも，誤ってフォーマットしてしまったディスクを復活させるSF (SafeFormat) とFR (FormatRecover) など，データ救済のためのツールがそろっている．また，逆にWIPEDISKやWIPEFILEのように，どのようなことをしても二度とデータを復活できないように消し去るためのツールも備わっている．

現在では，PC-9801版も発売されており，PC互換機以外でも使えるようになった．MS-DOSユーザーなら1本もっていても決して損はしないソフトである．

PROFILE



高井　護

最初に覚えたのは　8080のアセンブラ．当時は雑誌もなければプログラムを知っている人もあまりいなかったので，自分ひとりでコツコツと勉強したという．プログラムの参考にした本は，K&Rの第１版くらい．

プログラム　昔は趣味だったが，いまは完全に仕事．

WordMaster　はじめて触ったときの感動は，忘れられないとか．

趣味は？　オーディオとオートバイで，愛車はHONDA AX-1．

ースをまるごと取っておいても容量的にも余裕がある．

環境を知りつくせば非常に便利なものも，それを知らない人間にはなにもわからない．こうしたメンテナンス性を考慮して，開発環境はなるべくシンプルにしている．

少数精鋭主義　自由な雰囲気の中で自由な発想を

✚

メガソフトは少人数なので，特別に企画会議をもうけて製品企画をねるようなことはあまりしていない．つまり，プログラマ個人がアイデアを出し，それを商品に育て上げていくわけである．

プログラマのアイデアの段階では，それが実現可能なことなのかどうかもわからないことが多いので，まずプログラムを組んで実験してから製品開発にかかることもめずらしくない．

開発フローのようなものをきちんと決めて管理を強化するよりも，今の自由な雰囲気のほうがおもしろいものができると考えているので，自由な発想のできる今の仕事場を大事にしていきたいという．会社が大きくなると，どうしても個人の意見が反映されなくなり，管理強化が行われるため，今のままの自由で小さなソフトハウスでいたいというのがメガソフトの本音のようだ．

これからもメガソフトは，プログラマの右腕となるようなツール作りを続けていくということなので，おおいに期待したいところである．

MAX LINK　　OPTASM

プログラマの仕事場 | 8

Programmer's Working Space

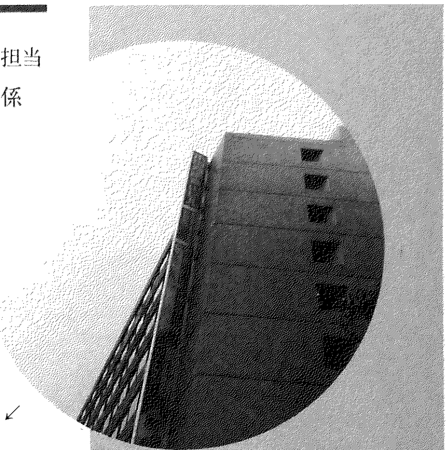
株式会社ボーランドジャパン

Turbo Cシリーズは優れた製品であるとともにボーランド社自身の開発ツールでもある

日本でTurbo C V1.5 PC-98用暫定版が発売されたのが，1988年1月のこと．その年のパソコンソフトウェアの売上ランキング第2位に輝く大ヒット商品となった．今日のC言語ブームも，Turbo Cの出現を抜きには語れないだろう．あれから3年の月日がたったいま，時代はC++へ移ろうとしている．各社が競ってC++のプロダクトの開発に取り組むなか，一歩先んじた形のボーランドジャパンが今後どのような展開を見せるか注目される．

1983年のTurbo Pascal発売以来，世界中にその名をしらしめた米国のボーランド社が，世界戦略の一環として日本進出をめざし，MSAと協力して設立したのがボーランドジャパンである．1989年４月のことである．両社の関係は，MSAがボーランド社と代理店契約を結んだ1983年12月にさかのぼる．ボーランド社の設立は1983年５月，Turbo Pascal発売が同年11月であるから，MSAの行動はかなりすばやいものであったといえる．翌年の４月にTurbo Pascal V2.0を日本語マニュアル付きで発売したのが，ボーランドの代理店として最初の活動である．以来MSAでは，次々とTurbo C，Turbo Pascalなどを日本語化して発売し，急速に売上を伸ばしていった．

現在では，ボーランドジャパンのほうで製品の日本語化などの開発を担当し，MSAはその販売元という関係になっている．そこで今回は，同社技術部長井原良平氏に，日本語の移植作業を中心とした開発環境について話をうかがった．



米国ボーランド・インターナショナル社が世界戦略の一環として日本に進出

✚

米国で生まれたソフトウェアということは，ターゲットマシンがIBM PCであるということだ．したがって，日本語の移植作業を行うためには，ソースマシンであるIBM PCとターゲットマシンである日本のパソコンの両方が必要ということになる．同社では，各自がIBM PCまたはAXマシンとPC-98という構成で，１人平均２，３台のパソコンを使っている．

ハードウェア構成はプロダクトの種

About Company

1989年４月設立．岡田純一社長．社員数約10名．

米国の大手ソフトウェアメーカーであるボーランドインターナショナル社が，日本のマイクロソフトウェアアソシエイツ（MSA）と協力して設立した同社の日本法人．出資はMSAが100%．ソフトウェアプロダクトの輸入，開発，販売を行っているが，おもにボーランド社のTurboシリーズなどの主力製品を日本語化して販売している．

Products

Turbo C++ V1.0
Turbo C++ V1.0 Professional

COLUMN

ボーランド・インターナショナル社

1983年５月設立．フィリップ・カーン社長．社員数約800名．
日本での通称は，ボーランド社．
本社所在地　米国カリフォルニア州スコット・バレー．
海外支店　フランス（パリ），イギリス（ロンドン），西ドイツなど．

1983年11月にTurbo Pascalを発売して以来，世界中をTurbo旋風に巻き込んだ米国のソフトウェアメーカー．Turbo C，Turbo Pascalなどの主力の言語製品のほか，多機能OAツールのSideKick，スプレッドシートのQuattro，簡易データベースのParadoxなどのアプリケーションを次々と発売し急成長してきた．現在では，世界中に多数のOEMメーカーやディーラーをもち，海外に数か所の支店をもつ大手ソフトウェアメーカーである．

社長のフィリップ・カーン氏は，Turbo Pascalを開発した技術者でもある．その優れた頭脳と独創的な発想を経営にもいかし，現在でも，同社の顔として世界を舞台に活躍中の人物である．

類によりさまざまで，メモリは1MBから10MB程度，ハードディスクは40MBから200MBぐらいとなっている．

開発環境にはLANが導入され，MS-DOSのネットワークソフトにはNetWareが使用されている．UNIXサーバーとして，VAX，NEWS，SUNなども使用し，これらはイーサネットで接続されている．またこのネットワークは，公衆回線で米国ボーランド社のインターナショナル開発グループとも結ばれている．

そのため，日本の開発担当者が必要なとき，即座に米国の担当者と情報のやり取りを行うことができ，開発効率を高めている．日本語移植のノウハウは機密事項で公開することはできないというが，基本的な方針として，米国でソフトウェアができあがってから移植に取りかかるのではなく，米国での開発段階から一緒に移植作業を行うことにしている．というのは，通常日本語化作業には短いもので4，5か月，長いものだと1年ぐらいかかるので，米国版の完成を待っていては遅すぎるとの判断からである．

"プロダクトの日本語化作業は米国での開発段階から一緒に行われる

プロダクトによっては，日本から技術者が米国に出向くこともある．出向かない場合は，米国で新規のベータバージョンができてから，日本でそれをチェックし，日本語化する仕様はこうだと先方に伝えて，双方で合意がなされてから変更にとりかかる．「できるだけ早く，日本のユーザーにボーランド社の製品を届ける」というのが，同社のポリシーの1つとなっている．Turbo Cが日本でも爆発的に売れた背景には，同社の素早いPC-98対応というものを忘れることができない．

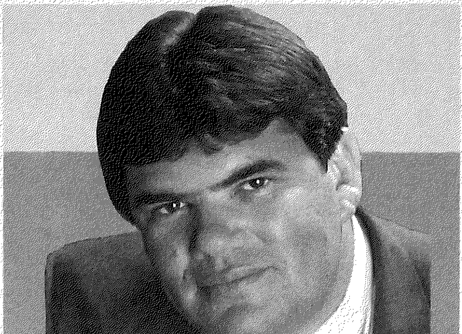
✚

開発ツール自体をあつかっている同社では，開発にはほとんどの場合，自

COLUMN

フィリップ・カーン

ボーランド・インターナショナル社長，フィリップ・カーンがコンピュータと最初に出会ったのは，チューリッヒ工科大学で数学を専攻していたときのことである．当時，彼は全くコンピュータに興味がなかったが，必修としてFORTRANかPascalのどちらかを選択しなければならなかった．当時FORTRANのクラスは400人，Pascalはたった6人だった．彼がPascalを選んだのは，学生数が少ないという理由だけだったという．そのときの教授が，Pascalの生みの親である天才ニクラウス・ワースであった．この出会いが彼の運命に大きな変化をもたらした．1971年，19歳のことである．



その後数年間，研究に研究を重ね，ようやく彼の理想とするPascalコンパイラが完成した．これが，のちのTurbo Pascalである．彼はそのPascalコンパイラを武器に，アメリカへビジネスを求めてやってきた．その後，さまざまな苦労の末，ベンチャーキャピタルの投資なしで，1983年5月にボーランド社を設立．4名のスタッフで，自動車修理工場の2階に事務所を構えた．しかし，同年11月発売のTurbo Pascalの爆発的ヒットによって，あっという間に，りっぱな自社ビルと約800名の社員をもつ，世界有数のソフトウェアメーカーに成長した．その経営手腕は，彼の技術者としての能力とともに，米国コンピュータ業界ではたいへん高い評価をえている．

社製品であるTurboシリーズを使っている．自社の市販のプロダクトを社内用に一部変更して使用するというケースも多い．ライブラリに関しても，同社のほとんどの製品はTurbo Cで書かれているので，市販のTurbo Cの標準ライブラリをそのまま使うことが多い．

「自分たちでいいと思ったものしか出さない．したがって，自分たちが使えないプロダクトは出さない」というのが，ボーランド社の創業以来の変わらぬ信念である．

たとえば，言語グループが新しく言語やツールを開発した場合，アプリケーショングループでは，それを自分たちのアプリケーション開発にいち早く利用しテストしてみる．その結果，もし使いにくい，あるいは利用価値が低いということであれば，さらにその言語に改良を加えていく．このように，まず社内で使用してみて，それが実用的で優れたものであると確信がもてた場合にのみ，商品化に踏み切ることにしている．こうした姿勢こそ，今日までボーランド社の製品が，使いがってのよさでユーザーに評価されてきた由縁であろう．

開発ツールはほとんどがTurboシリーズのもの
優れた製品を自社で使うのは当然だ

Turbo C V2.0
Turbo Debugger & Tools V2.01
Turbo Pascal V5.5
Turbo Pascal V5.5 Professional 2nd Edition
Turbo Pascal 5.0
※販売元はMSA

そのほかの開発ツールとしては，Turbo Debuggerでも大きすぎて追えないようなもののために，常駐型デバッガを作ったり，各自が必要に応じて自由に作成して使っているという．

✚

今後のプロダクトとして，同社がとくに力を入れているのは，言語ではTurbo C++，アプリケーションでは

COLUMN

Turbo C++

C++は，AT&Tベル研究所がCの後継言語として開発した，オブジェクト指向型プログラミング言語．Turbo C++は，AT&TのC++ 2.0とANSI Cをサポートする高速C++コンパイラの日本語版．EMSとプロテクトメモリのサポート，ヘッダファイルの属性の自動チェックなどのコンパイラ機能のほか，マルチウィンドウやマウスにより，効率のよい開発環境を提供する．Turbo C++ Professionalは，Turbo C++に，Turbo Debugger，Turbo Profiler，Turbo Assemblerなどの開発支援ツールをセットにしたもの．

Turbo Pascal 5.5

Turbo Pascal 5.5は，従来のTurbo Pascalの機能に，オブジェクト指向プログラミング機能を追加した高速Pascalコンパイラの日本語版．Turbo Pascal V4.0，5.0との高い互換性をもち，毎分34,000行のコンパイル，64KBをこえる実行プログラムの作成が可能．

Turbo Pasclaは，Ver.1.0以来，圧倒的な支持をえており，パソコン上のPascalコンパイラとして標準的な存在である．


米国ボーランド社の本社

Paradoxである．さらに，MS-Windows 3.0にも積極的に対応していく予定である．

日本ではまだあまり知られていないが，Paradoxは，現在米国ではたいへん売れているデータベースパッケージで，高速で使いやすいという評価を得ている．米国ボーランド社では，ParadoxをdBASEに対抗するデータベースソフトとして，大々的な広告とともに，売り込み攻勢をはかっている．現在，Paradoxの日本語化が進行中とのことで，今後日本市場でどのような広がりをみせるか，注目したいソフトウェアの1つである．

このほかにも，Quattro Proというスプレッドシートが，Lotus 1-2-3に対抗するものとして米国では話題を呼んでいるが，こちらのほうの日本語化は，現在未着手とのことだ．しかし，米国ボーランド社には，「ボーランドの製品はどこの国でも動かす」という意向があるため，いずれは日本語版も登場することになるだろう．

今後それらのソフトが，日本の市場で不動の座を守るLotus 1-2-3やdBASEを相手に，どこまで健闘をみせるのか，興味深いところである．

Turbo C++とParadoxをこれからの展開の軸に

PROFILE

各プロダクトの開発担当者は全員TK-80の元ユーザーというマニアたち

各プロダクトのプログラマは，それぞれユニークなキャラクターの持ち主であるが，ただ1つ共通しているのは，全員が紛れもないパソコンマニアであるということだ．インタビューした4人全員が，あのワンボードマイコンTK-80のユーザーであったというのには驚かされた．



ボーランドジャパン技術部長
井原　良平 氏

1年の約半分は米国生活

年齢　32歳．

趣味？　音楽演奏，写真，登山など．

大学時代　富士通のFM-8を買い，そのおもしろさに魅せられ，その1週間後からMSAでアルバイトをはじめる．以来10年，表計算ソフトSuperCalcの日本語版開発などを経て現在に至る．

米国ボーランド社との往復　1年の約半分を米国ですごす．もともとはプログラマであるが，現在は開発管理の仕事が中心で，プログラムはときたまスタッフの手伝いをする程度．個性の強い技術者たちを，それぞれのパワーを落とさずにいかにまとめていくかが，常に悩みの種である．

PROFILE

有澤雄志

技術部．Turbo Pascalなど言語関連担当．25歳．

趣味？ 山登り，ハイキング，神保町散策．

中学時代 TK-80を使ったのが最初．

開発経験言語 C，Pascal，アセンブラ．

自宅のパソコン ①NEC PC100(2DD-FD×2)／メモリ 256KB ②CPM180 (2HD-FD×3，8"FDD×1)／メモリ 256KB

会社のパソコン ①EPSON 386-LS (80MBHDD)／メモリ 640KB+2MB ②SONY QuaterL (40MB HDD)／メモリ 640KB+1MB ③NEC PC9801RX (80MBHDD)／メモリ 640KB+4MB

田中　勉

技術部主任．Paradoxなどアプリケーション担当．32歳．

趣味？ 音楽観賞．

大学の研究室で TK-80を使って制御系を経験したのが最初．

漢字がなかったら 開発がどれほど楽だったか……．

開発経験言語 C，アセンブラ．

自宅のパソコン ①MAC SE (45MBHDD)／メモリ 2.5MB

会社のパソコン ①NEC PC9801VX(40MBHDD)／メモリ 3MB ②SONY QuarterL (130MBHDD)／メモリ 3MB

田川洋一

技術部．Turbo C++など言語関連担当．24歳．

趣味？ コンピュータ以外にはなし．

12歳のとき はじめてTK-80を使った．18歳のときから仕事でプログラムを組むようになる．24歳にして，すでに人生の半分をプログラマとして送っている．

開発経験言語 アセンブラ，C

自宅のパソコン ①MAC SE (20MBHDD)／メモリ 4MB

会社のパソコン ①NEC PC-H98 (100MB+200MBHDD)／メモリ 9MB ②SONY QuaterL (40MBHDD)／メモリ 1MB

青木雅祐

技術部．Turbo Debugger担当．27歳．

中学時代に TK-80をはじめて使った．

開発経験言語 BASIC，FORTRAN，C，アセンブラ

自宅のパソコン ①MAC SE (40MBHDD)／メモリ 4.5MB ②NEC PC9801VM／メモリ 1.5MB

会社のパソコン ①NEC PC-H98 MODEL70 (180MBHDD)／メモリ 9.5MB ②SONY QuaterL (40MBHDD)／メモリ 1.5MB ③EPSON PC-386LS／メモリ 1.5MB



プログラマの仕事場 | 9

PW
Programmer's Working Space

株式会社ジャストシステム

ヒットメーカーの原動力はインテリジェントビルに集う120人のプログラマ

日本語ワープロソフトのベストセラー「一太郎」の開発元である株式会社ジャストシステムは，「踊るあほうに，見るあほう……」の阿波踊りで有名な，四国は徳島にその本社をおく。徳島市郊外にそびえるインテリジェントビル内では，約120人のプログラマたちが，一太郎をはじめとするさまざまなアプリケーションの次なるバージョンや新たな製品の開発に従事している。
このジャストシステム本社を訪ね，ジャストシステムにおけるプログラマの仕事場について取材した。

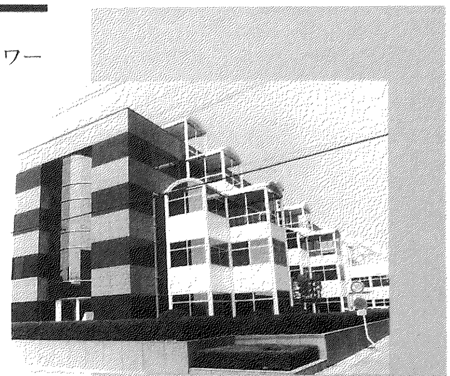
オンラインソフト情報
TODAY.EXE(TODAYV28.LZH)　今日が何の日か教えてくれるユーティリティ．記念日，雑誌の発売日などのほか登録も可能

株式会社ジャストシステムは，パソコン用日本語ワープロソフトのベストセラー「一太郎」の開発元として，広くその名を知られている．

いまでこそ社員数400人以上，年商89億円（90年4月期）を誇る業界有数の企業に成長したものの，1979年の創業時には浮川和宣社長と浮川初子専務の2人だけの小さな会社だった．当時は，オフコンの販売とそれに伴うソフトの開発がおもな業務だったが，このころからすでにコンピュータによる日本語処理の研究を進め，日本語に強いジャストシステムという評価を受け↙

1985年には，自社ブランド初のワープロソフトjX-WORD太郎を発売．わずか半年後，太郎はバージョンアップし，一太郎となった．その後，1987年にはベクトルデータを扱う図形プロセッサ「花子」を，1988年にはビットマップデータを扱うイメージライタ「シルエット」および日英ワープロソフトの「デュエット」を，1990年には通信ソフト「Let's talkかたろう」を発売している．

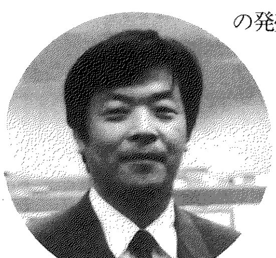
ジャストシステムでは，現在，一太郎をはじめとする既存アプリケーシ↙



ジャストシステムの過去，そして現在 オフコンからパソコンへ，一太郎からDTPへ

ていた．

“漢字が使えるパソコン”PC-8801の発売をきっかけに，パソコン用の日本語処理システムの研究をはじめ，16ビットMS-DOSマシンの登場とともに本格的参入を果たした．1983年には，ジャストシステム初のパソコン用ワープロソフトJS-WORDが，NEC PC-100に標準バンドリングされた．このとき浮川社長は，「あなたに日本をまかせますよ」といわれたようなショックを受けたという．



浮川和宣社長

ョンのネットワーク対応化と，DTPシステムの開発が進められている．オフィスコンピュータの時代から日本語と漢字を追い続けてパソコンにたどりつき，さらに，パソコンにおける日本語システムを追及した結果DTPにたどりついたというのが，いまのジャストシステムの姿のようだ．

✚

現在，徳島市郊外の本社ビルには，約120人のプログラマが開発に従事している．本社ビルの室内照明には，コンピュータのディスプレイ

About Company

現在のパソコン状況が成立する過程でジャストシステム，とくにその製品である「一太郎」「花子」がはたした役割は大きい．このパッケージの成功は，徳島の1ソフトハウスを全国的な知名度をもつ存在にまでした．徳島市内に2つの本社ビルが完成したいま，総社員数は400名をこえている．

Location

徳島市に本社をおき，港区芝に東京支社がある．本社から打ち合わせなどで東京にでることも少なくないというが，さほど不便は感じないという．徳島・羽田間の飛行機を利用すれば，本社を出てから2時間ほどで着いてしまうからだ．時間的には東京郊外から都心に出るのとあまり変わりがない．

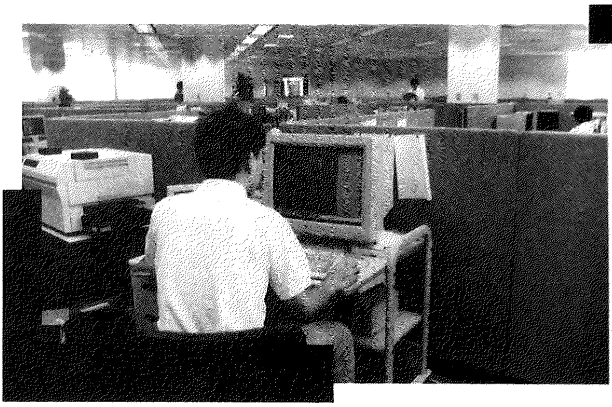
表1：JUSTSYSTEM HISTORY

1979.7	徳島市でジャストシステム創立
1983.10	PC-100対応ワープロソフト「JS-WORD」開発
1984.5	日本語FEP「KTIS2」開発
1984.11	IBM JX用ワープロソフト「jX-WORD」発売
1985.2	PC-9800対応ワープロソフト「jX-WORD太郎」発売
1985.8	PC-9800対応ワープロソフト「一太郎」発売
1986.5	「一太郎Ver.2」発売
1986.12	PC-98LT用ワープロソフト「サスケLT」発売
1987.3	図形プロセッサ「花子」発売
1987.6	「一太郎Ver.3」発売
1988.5	イメージライタ「シルエット」発売
1988.6	日英ワープロソフト「duet」発売
1989.4	「一太郎Ver.4」発売
1989.9	「花子Ver.2」発売
1989.10	「一太郎dash」発売

に反射しない特殊なものが用いられ，床下にはSUNのワークステーションをサーバーとしたLANに接続するためのイーサネットケーブルが張り巡らされている。

開発室というよりは，開発フロアと呼ぶべき広いスペースが整然とパーティションによって区分けされ，そのパーティションのなかで数台のパソコンに取り巻かれるようにして，プログラマたちは開発を進めている。

開発に使用されるマシンのほとんどはNECのPC-9800シリーズで，プログラマ1人に対して約4.5台が設置されている。これらのマシンはすべてLANで接続されており，プロジェクトの内容に応じてMS-DOSマシンとして使用することも，UNIXの端末として使用することもできる。また，これらすべてのマシンにハードディスクとジャストシステム製のEMSメモリボードが搭載されているそうだ。



パーティション内に設置されたもの以外にも，フロアの要所要所に複数のパソコンがおかれており，必要があればいつでも利用できるようになっている。数は少ないものの，MacintoshやNeXTなども研究用に準備されている。

開発マシンは1人平均4.5台 ユーティリティ類は自社開発

プログラム開発はC言語とアセンブラで行われ，コンパイラはMS-C，アセンブラはMASMを使用している。エディタは，全員がMIFESを利用している。これは，ジャストシステムの標準エディタとして会社側が用意したものだ。

エディタ以外のリンカ，デバッガなどの開発ツールやライブラリなどは，現在のジャストシステムの開発環境に即したものを自社で開発して利用することが多い。

✚

ジャストシステムにおける製品開発のプロジェクトは，いくつかの部署で分散して行われる。まず，プロジェク

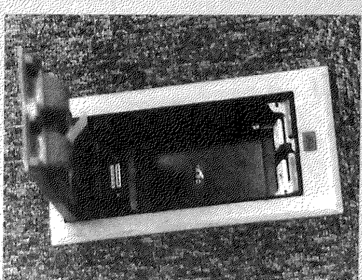
COLUMN

イーサネット　UNIXワークステーションの標準ネットワーク

イーサネット（Ethernet）は，米国のゼロックス，インテル，DECの3社が共同開発したLAN（Local Area Network）である。

具体的には，一般にイーサネットケーブルと呼ばれる黄色の同軸ケーブルとトランシーバ，トランシーバケーブルを用いてワークステーションやパソコンを接続する。トランシーバは，データ送信の際に同軸ケーブルを監視し，同軸ケーブルが空いているときにデータを送信する。データ受信の際には，トランシーバが同軸ケーブル上を通るすべてのデータを受信してしまうため，トランシーバの上位のインターフェースがこれを取捨選択することになる。

イーサネットは，比較的安価で簡単にワークステーションや端末を接続できるため，UNIXワークステーションを用いたネットワークで広く用いられている。



床のコネクタにつなぐだけでLANシステムに

ト推進という部署が，営業の意見およびユーザーサポートの意見を社外からの要求項目としてまとめる．次に部長会で話を進め，さらに開発のリーダー，サブリーダー，プロジェクト推進の人間を含めて話し合いを進める．こうしておおまかなスペックが決まると，実際にコーディングを行うプログラマを集めてスペックの説明が行われる．ここでプログラマ自身が話し合いに参加して，仕様決定に加わることになる．

基本的にはトップダウンで開発が進められるわけだが，常に話し合いの場において，下からのフィードバックがなされるため，現場で働くプログラマの意見も十分製品開発に生かされるということである．

開発進行は「プロ推」グループが担当 製品開発に平行してマニュアルを作成

この製品開発フローの中でとくにユニークなのが，プロジェクト推進だろう．この部署は，平均年齢25歳の女性約20名で構成され，ソフト開発の進行管理とマニュアルライティングを行う．

「プロ推」と呼ばれるこの部署のメンバーは，プロジェクトごとに1人から6人が割り当てられ，企画段階の仕様書の決定に参加するとともに，コーディングが開始されてからは担当プログラマと密接なミーティングをもってスケジュール管理を行う．それと同時に製品マニュアルを執筆するのである．

つまり，ジャストシステムでは，製品の開発とマニュアルの作成が同時進行で行われているわけである．

プログラマの多くが，ジャストシステムにきてからプログラミングを身につけたということからもわかるように，社内教育にはかなり力を入れている．

4月に入社してきた新入社員は，まず本棚1段分ほどもある参考文献を手渡され，その日から3か月間にわたる集中研修で，MS-DOS，C言語，アセンブラを徹底的にたたき込まれる．

研修期間中は，毎日ベテランプログラマの講義を受け，さらにレポートとして課題プログラムを書いて提出することが義務づけられる．こうしてプログラミングの基本を身につけたところで現場に配属され，実際の製品開発にかかわりながら，さらに先輩プログラマからプログラミングのノウハウを学んでいくことになる．

プログラミング教育においては，トリッキーなプログラムをせず，なるべくスタンダードなプログラミングを行うように指導している．このため，とくにコーディングのフォーマットなどを決めなくても，わかりにくいソースが出てくるようなことはほとんどないという．

徹底した社内教育体制のもとで 新入社員は3カ月間の集中研修を受ける

また，ジャストシステムにおいては，プログラマだけではなく総務に配属される人間であっても，必ずC言語の研修を受けさせられる．これは，ソフトウェアビジネスに関わる限り，どのような部署で働いていても，ある程度プログラミングができなければ話にならないというジャストシステムの主張からきている．こうした社内教育は，ジャストシステムの社員が10人，20人だったころから続けられている．

現在ジャストシステムでは，各アプリケーションのLAN対応，DTPソフトのシリーズ化などをはじめとする30以上のプロジェクトが進められている．

浮川社長によると，ジャストシステムがこれから進めようとしているプロジェクトは，とても2～3人のプログラマでできるようなものではないので，120人のプログラマを擁するいまも，まだまだプログラマがたりない．

ジャストシステムに所属するプログラマのうち約4分の3が県内出身者，4分の1が県外出身者だが，優秀な人材はのどから手が出るほど欲しいので，ぜひ徳島にきて欲しいと浮川社長は語っていた．

プログラマの仕事場 10

Programmer's Working Space

株式会社大塚システム研究所

LANを導入した開発環境には パソコンソフトウェア開発の 創造性が息づいている

大規模な社内通信ネットワークの構築，開発環境へのLAN導入など，大塚システム研究所はシステム開発の効率化を実現してきた．同社のインテリジェントビルでは，ビル全体を管理するメイン・ホストを設置し，その下に各階のホストマシンをつなげている．さらに各階では開発プロジェクトごとに異なるホストとLANを導入し，サーバーを中心として効率的に開発を進めている．開発環境におけるLAN導入はこれからますます増えていくものと思われるが，その実態についてレポートした．お話は同社開発部データベース開発課係長心得花島輝彦氏にうかがった．

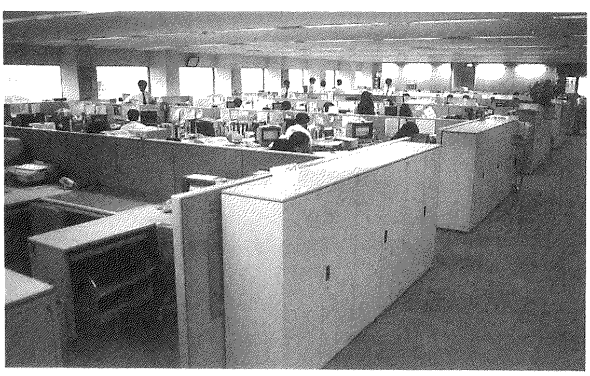
同社では，パッケージの場合は企画の段階から社内で行っている．そのために，アンケート調査，ユーザーに直接話を聞くなど，マーケット調査も行う．営業担当者やセンターショップなどから情報を得ることもある．

花島氏の所属する開発部は，パソコンアプリケーションの開発を行っている．開発部は，データベース，ワープロ，CAD，ネットワーク，I/O調査，基本OS，フィールドサポートの7グループに分かれている．製品開発は，各グループ内でプロジェクトを組み，構想を練りそれを開発会議に提出することからはじまる．そこで上司の承認が得られた後，人員の選出が行われ，開発作業がスタートする．基本設計の段階は数名で，プログラム開発に取りかかる段階ではもっと人数を増やすなど，工程により開発グループの人数は異なる．プロジェクトのメンバーにはSEとプログラマの区別はなく，たいていの場合，1人がその両方を兼ねることになる．

それは，パソコン上のソフトウェア開発の特色であると考えられるだろう．商品企画，ソフトウェア設計，コーディング，デバッグなど，各段階で人員の増減はあるものの，中心となる人物は全工程をとおしてプロジェクトにかかわることになる．SEからプログラマへ，仕様書だけが手渡されるような世界ではない．

開発チームにはSEとプログラマの区別はない 基本設計に関わったものが最後まで担当する

「基本設計にかかわったものが最後までやる」というのが，効率と品質を考慮した上でのポリシーとなっているようだ．それぞれの工程をとおして牽引できる力をもったものが，プロジェクトリーダーとなる．1グループの人数は3，4名から20数名，開発期間も3か月から2年位とプロダクトの種類によりさまざまだが，1グループで平均2，3のプロダクトを並行して担当しているので，常に20前後のプロダクトが動いていることになる．



＋

開発部のホストは，当初DECのVAXであったが，現在はヒューレッドパッカード社のUNIXマシンを採用している．開発部ではホストの下をグループ単位にイーサネットで接続し，グループ用のホストマシンを設置している．グループによりホストマシンとLANの種類は異なり，たとえば，ワープログループでは大容量ディスク付きのPC-9800シリーズパソコ



About Company

株式会社大塚システム研究所．昭和59年設立．大塚　実社長．社員数約300名．

大塚商会の技術開発部門として，昭和59年に大塚システムエンジニアリングの社名で設立．昭和63年に大塚商会のインテリジェントビルの完成と同時に，社名を大塚システム研究所に変更し，同ビルに移転した．パソコンおよびオフコンのアプリケーションパッケージの作成，オーダーメイドソフトの開発などを業務としている．

Products

MS-Windows対応データベースソフト
　PAL-WINDOWS
MS-Windows対応CADソフト
　EXPERT-CAD
日本語ワードプロッセッサ
　オーロラエース
簡易言語型データベース
　PC-PALスーパープラス
　PC-PAL Jr.
2次元汎用CADシステム　PC-CAD
※全て大塚商会から販売されている．

LAN開発環境では共通のルールのもとで開発を進めて行くことが重要となる

ンにスターラン，データベースグループでは3B2（AT&T社製）にスターラン，CADグループではEWS4800（NEC社製）にNet/Oneを使っている．その下に，1人平均1台のパソコン（おもにPC-98）がそれぞれのホストに接続されている．花島氏がリーダーを務めるデータベース・グループでは，3B2の144MBのハードディスクだけでは足りなくなったので，現在2台目のサーバーとして，その下に計430MBのハードディスク付きのPC-H98を接続している．さらにその2台のホストの下に，PC-98が30台近くも接続されている．3B2にはライブラリ，リンクする環境だけがおかれ，PC-H98にソースがおかれている．各自が必要なものをホストから自分のパソコンにダウンロードして，そこで修正しライブラリを更新するという仕組みになっている．データベースグループは，開発部の中でもネットワーク使用頻度が一番高いというが，そのLANシステムの構成は図1のようになっている．

✚

LANを使用した開発環境では，開発中のソース，ライブラリは全てサーバー上にのせ，全員が共有してそれらを使用するので，リンクもその中で行われている．したがって，サーバー上の共通ライブラリ管理には，図2のように特別なルールが設けられている．共通のライブラリとして認められると，正式なものとしてリンクされる．

✚

また，コンパイラ，リンカなどのツール類は，ネットワークのものを直接使うと多少時間がかかるので，個人が自分のハードディスクにダウンロードしてから使うことが多い．そのため，1つのプロダクトがはじまるときやそれらのツールに変更があった場合，常にサーバー上のものを更新して統一し，全員に通達を回すことをルールとしている．このように全員が共通のルールのもとで開発を進めることによって，LANの機能が一層活かされている．

LAN導入前の開発環境での最大の問題点は，共通ライブラリの更新ミス

図1

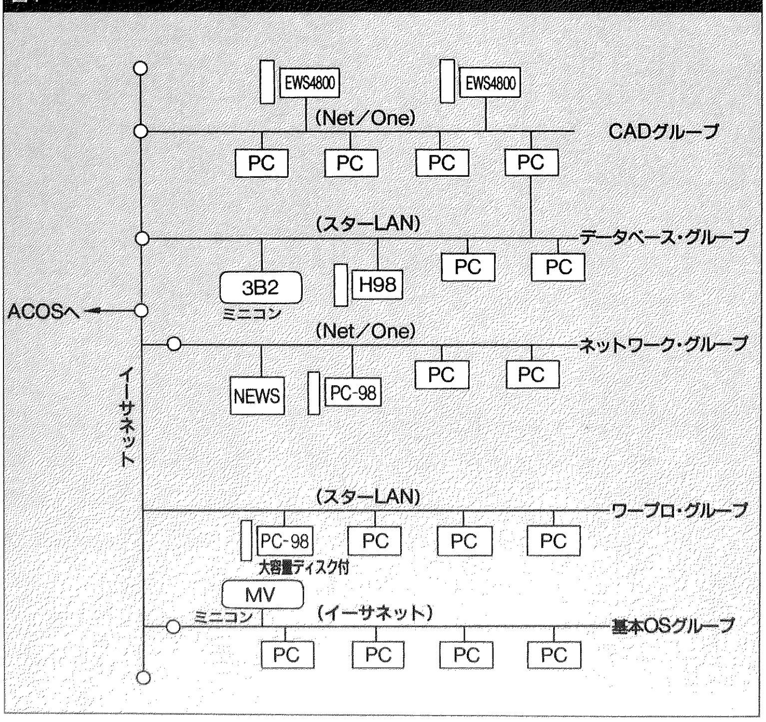
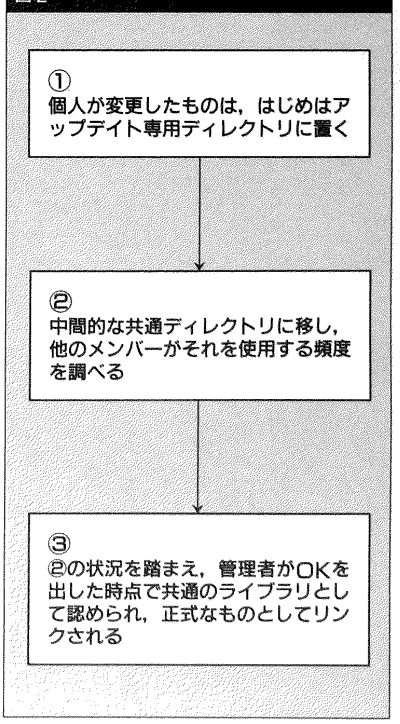


図2



だった．当時は個人が自分のハードディスクで修正を加えたものを，ほかの共通のリンク専用パソコンにもってきて，そこでライブラリのリンクを行っていた．そのときは，共通リンク専用マシンのところに「今日の更新分」と書いたフロッピーを毎日おいておき，各自がそれで自分のパソコンのライブラリを更新するようにしていた．

しかし，実際にはよく忘れてしまうことがあり，固有シーケンス番号が違ったままテストし，そこでは動いたものの，最新のものにもってくると動かないという問題がよくあった．プロダクトに関わる人数が5，6名の場合はそれもあまり気にならなかったが，やはり10名以上のプロダクトになると開発工数に関わる重要な問題となり，なんとかしてそれを解決したいと思っていたという．

LAN導入の最大メリットはライブラリの更新ミスが皆無になったこと

そんなときに，スターランをテストしてみてくれないかという話がきて，それがLAN導入のきっかけとなった．導入に先立って，簡単にライブラリの更新ができないようにするなどいくつかのルールを定め，導入と同時にそれらを全員で実施したところ，見事に更新ミスが皆無になった．当然，それによって開発工数の無駄がなくなり，導入前の問題が解決したというわけである．

LAN導入によって共通ライブラリの管理という点では成功をおさめ，また膨大なライブラリという資産を残すことができたが，現在のLANにはまだいくつかの課題がある．そのひとつに，ネットワーク上のセキュリティの問題があげられる．

現在は，MS-Networksを使っている制約上，システムのセキュリティは低い．そのため，機会があったらシステムのセキュリティをあげようと考えているそうだ．そのひとつの契機となりそうなのが，開発システムへのWindows 3.0の導入だ．Windows 3.0は，ネットワーク対応なのであり，その日本語版がでた時点でNetWareなどによって新たにネットワークを組み直すことも考えている．そうなれば，ネットウェアの機能によりセキュリティも強化されるだろう．

✚

ワープログループで使っているパソコンは，同社のおもなターゲットマシ

表■PALの変遷

製品名 リリース年月	対応機種	開発言語	備考
PC-PAL Ⅰ (1981年9月)	PC-8001	N-BASICとマシン語	PALシリーズの誕生
PC-PAL Ⅱ (1982年7月)	PC-8001 PC-8801	N-BASICとマシン語	機能強化
PC-PAL日本語 (1982年12月)	PC-9801	N86-BASICとアセンブラ	漢字の使用を可能
PC-PAL日本語高速版 (1983年2月)	PC-9801	N86-BASICとアセンブラ	高速化
PC-PAL日本語辞書付 (1983年10月)	PC-9801	N86-BASICとアセンブラ	日本語入力機能向上
PC-PAL日本語辞書付 Ver.2 (1984年4月)	PC-9801	N86-BASICとアセンブラ	機能強化
PC-PALスーパー (1985年9月)	PC-9801 PC-98XA	N86-BASIC，アセンブラ C言語	MS-DOS対応
PC-PALスーパー Ver.2 (1986年8月)	PC-9801 PC-98XAなど	C言語(Lattice-C)	機能強化とC言語化
PC-PALスーパープラス (1988年9月)	PC-9801	C言語(Lattice-C)	ネットワーク対応
PC-PAL Jr. リリース (1990年2月)	PC-9801	C言語(Lattice-C)	ノートパソコン対応
PAL-Windows (1990年9月)	PC-9801	C言語(MS-C)	MS-Windows対応

開発用パソコンの主流はCPU80386マシン コンパイラはMS-C

ンであるPC-9801が多い．MS-Windowsをはじめてからはテストのため，IBMなどの機種も増えてきた．データベースグループを含め開発部ではPC-9801RAが多いが，CADグループなどではPC-H98やPC-9801RLなどハイレゾリューションの機種も利用．全体として，最近ではCPUに80386を搭載したものが主流になっている．

ディスクは内蔵ハードディスクを使用しているところが多いが，マシン構成が変わりやすい部署では外付けのものを使用している．ハードウェアの増設状況はプロダクトの種類によりさまざまだ．Windows関連の部署では，EMS用に4MBのメモリと40MBのディスクが増設されている．それ以外のところでは，RAMディスクとキャッシュを使用し，ソースをRAMディスクに入れコンパイルしている．

開発部全体を見てみると，半数ぐらいがハードウェアを増設している．メモリはNECの純正のものだけでなく，メルコやI・Oデータ機器のものもテストのために使用している．ドライバソフトもNEC以外のMELWARE，IOS-10IIやMemory-pro386なども使用している．これには，ユーザーの利用するさまざまな環境でテストするためという理由もある．

OSはMS-DOSが大半を占める．コンパイラはおもにMS-Cだが，一部で旧ソフトウェアのメンテナンス用にLattice-Cも使用している．エディタは各自の好みで自由に選択している．

そのほか，WINDOWS ToolKitを使っているのはもちろんだが，こまかなものでは，必要に応じて社員が開発したツールを共有の資産として使うこともある．

✚

花島氏がパソコンの開発部門に転属したのは1982年のこと．はじめての担当したのはPC-PAL日本語，N86-BASICとアセンブラで開発した．

「PC-PAL日本語がBASICをメインに使用しているため，BASICの特徴であるガベージコレクションという内部のエリアの再配置作業が発生し，PALの動作中やプログラムの切り替え時に一時的に止まってしまうという現象が発生した．これを修正したのがPC-PAL日本語高速版だ．このガベージコレクションはBASICの性格上どうしても発生して

PROFILE



花島輝彦

開発部データベース開発課係長心得．

年齢　33歳．

趣味　映画観賞，写真，ゴルフ．

PALシリーズ　オフコン関連の営業を経て，自ら希望しパソコン部門へ．最初は，PAL日本語を担当．以来PALシリーズの開発に携わり，最近では「PAL-WINDOWSのプロジェクト」リーダーを務めている．

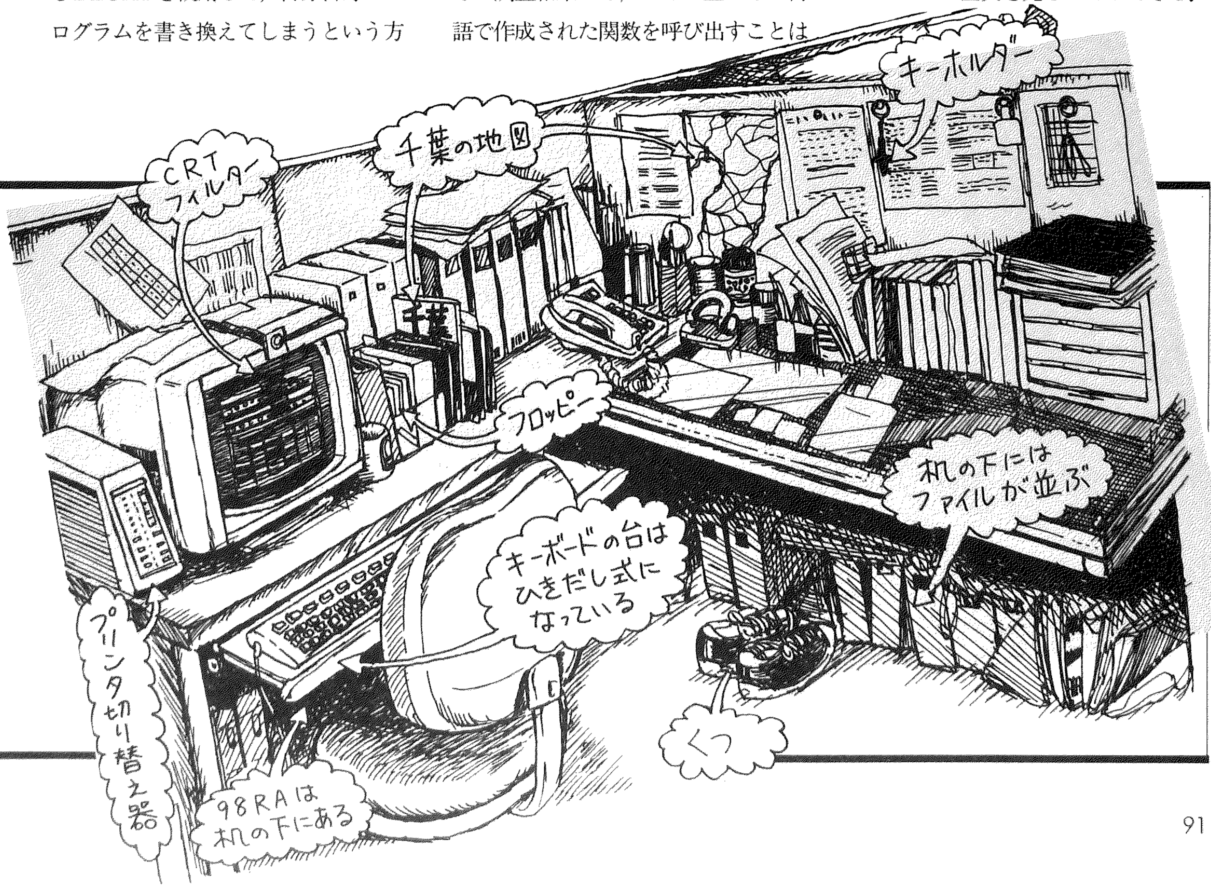
しまうもので，メーカーなどに相談しても，避けられないということだった．しかし，この問題を解決しなければ『ユーザーの納得のいくものにならない』と考え，BASICの調査，研究を行った．約1か月が過ぎたころ，先輩が『わかったぞ！　LSET，RSETだ』と叫んだ．なんと文字列操作を行うときにバッファ操作用のLSET，RSETを通常の文字列に使用することで実行中のガベージコレクションが発生しなくなるのだった．また，プログラムの切り替え時(CHAIN命令)に発生するガベージコレクションについても，同様にBASICを共通部とコマンド部に分割し，コマンド部のプログラムをインタプリタとして読込まず，マシン語の読込みであるBLOADを使用して，自分自身のプログラムを書き換えてしまうという方法で回避した」．

はじめてC言語を一部採用したのが，PC-PALスーパーだった．「MS-DOS上で動作することが条件だったため，今までのDISK-BASIC上での開発とはまったく違う開発環境だった．当時，MS-DOS上で開発を行うとしたらC言語が有力だったが，C言語を組める開発人員が少なかったため，3つの言語の混在で作成することになった．しかし，BASICとC言語を混在して利用する技術など当時はまったくなく，まずはMS-DOS上のBASICとC言語の調査からはじめた．その調査結果から，BASIC上からC言語で作成された関数を呼び出すことはできなくても，BASICからアセンブラを呼出し，アセンブラからC言語の関数を呼出すという方法を見つけだすことに成功し，採用した」と，花島氏は話す．

BASICとアセンブラからLattice-C そしてMS-C中心の時代へ

開発を全てCで行なうことになったのは，PC-PALスーパー Ver. 2からだ．「NEC以外の機種への移植の話がもちあがり，移植するためにBASICの利用を断念し，C言語とアセンブラに変更し，処理スピードのアップとプログラムサイズの縮小を行った」．

同氏のプログラマとしての歴史は，そのままPALの歴史でもある．またそれは，パソコンの歴史の多くの点(OS，開発言語，使用機種など）で重なっており，そこに日本のパソコンアプリケーションの歴史を見ることができる．



プログラマの仕事場 11

Programmer's Working Space

小高輝真(フリープログラマ)

ハードの性能を生かす アセンブラによる開発システム 押入にはハードウェア工房も

小高輝真氏は，雑誌・書籍などでライターとしても活躍している若きフリープログラマである．ソフトはむろんのこと，ハードウェアに関する深い知識をもち，アセンブラを用いてハードの性能を生かしたプログラム開発を行っている．

パソコン通信においても，その知識を生かして日経MIX PC-9801会議の議長を務め，悩めるPC-9801プログラマの救い主となっている．

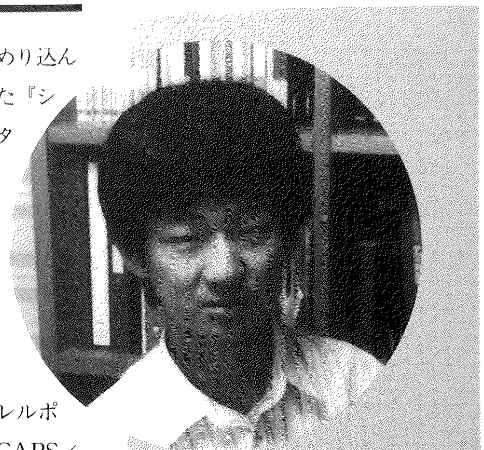
千葉県流山市の自宅におじゃまして，コンピュータとの出会いから，情報入手の方法まで，プログラミングライフを語ってもらった．

小高氏がはじめて触れたコンピュータは，中学校3年生のときに買ったPC-1211というBASICが走るSHARPのポケコンだった．このポケコンで，コンピュータがどんなものかを知ったのである．小学校の6年生のころから電気工作のようなことをやっていたので，コンピュータの世界にもすんなり入っていけたということだ．

高校1年生のころに8ビットの新しいポケコン（PC-1500）を購入し，ますますポケコンの世界にのめり込んでいったが，のちに出版された『シャープポケットコンピュータPC-1500機械語の手引』という本を入手したのが決定打となった．この本を手垢で真っ黒になるまで一生懸命繰り返し読んで，機械語を完全にマスターしたのである．

これ以後，ポケコンにパラレルポートをつけたり，かなキーがCAPSロックに切り替わるようにしたり，2KBほどRAMを増設したりと，ポケコンのハードウェアの改造を行い，さらに，増設したパラレルポートに接続するD/Aコンバータやミュージックジェネレータボードなどを作成した．

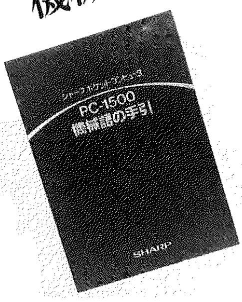
このポケコンの改造の話を雑誌『プロセッサ』に投稿したのがきっかけで，パソコン雑誌に記事を書くようになり，いまでは技術評論社のマニュアルや書籍などの原稿を執筆している．



Profile

小高輝真
小学校の電気工作から中学生のポケコン，さらにパソコンと順調にバージョンアップ（?）を重ねてきたパソコン青年．パソコン通信では，日経MIXのPC-9801会議の議長kodakoda氏として知られる．パソコン以外の趣味は音楽ということだが，ほんとうの趣味は宴会ではないかとの噂もある．

ポケコンでプログラミングをマスター 機械語とハードウェアの基礎を習得した



『PC-1500機械語の手引』この本が小高氏の原点

COLUMN

make (MS-DOS SOFTWARE TOOLS 3)

makeは，もともとはUNIX上で使用されていたツールで，プログラム開発にともなうコンパイル，リンク，ファイル管理などのコマンド実行を半自動化するためのコマンドである．プログラマは，makeによってプログラム開発時の雑事から解放され，ソースファイルのエディットに集中することができるようになる．

makeは，ターゲットとなるオブジェクトファイルを作成するのに必要なソースファイルが，オブジェクトファイルが作成されたときよりも後で変更された場合に，半自動的にこれらの変更されたソースファイルを再コンパイル・再リンクして，新たなオブジェクトファイルを生成する．

makeが参照するメイクファイルは，プログラムのソースファイルと共に保管することによって，そのプログラムの開発者以外の人に対して，そのプログラムのコンパイル方法に関するドキュメントとしての役割を果たす．

現在では，MS-DOS上で動作するほとんどのCコンパイラにmakeが付属してくるようになっている．また，makeを単体を製品としたものや，フリーソフトウェアのmakeなども存在している．

1984年に待望のパソコンPC-9801Fを購入，以後PC-9801VX21，PC-9801RA21とPC-9801シリーズを使い続けている.

現在のメインマシンはPC-9801RA21で，これにはICMのSRC131という131MBのハードディスクとI・Oデータの4MBのEMSボードを接続している．また，PC-9801VX21にはジャストシステムの2MBのEMSボードが入っている．このVX21には，バンクメモリにアクセスしたときに点灯するアクセスランプが付けられているが，これは小高氏が自分で設計・改造したものだ.

開発システムは9801を3台活用「椅子も環境のうち」というこだわりも

また，RA21には3.5インチの外部ドライブが接続されているが，これも小高氏が秋葉原で買ってきた裸のドライブを自分で取りつけたもので，RA21内部の基板に直接配線することで2HD/2DDの自動切り換えを可能にしている．これら以外にも，サウンドボードとSTARFAXが備えられている.

プリンタは，日本語レーザープリンタPC-PR406Mを使っている．普通のドットインパクトプリンタでは打ち出すときの音がうるさすぎるし，プリントアウトのスピードも遅いため，とてもソースリストなどをプリントアウトする気にはなれないが，レーザープリンタは静かで，しかも縮小文字で打ち出すとA4用紙1枚にソースがたくさん入って見やすいので気に入っているということだ.

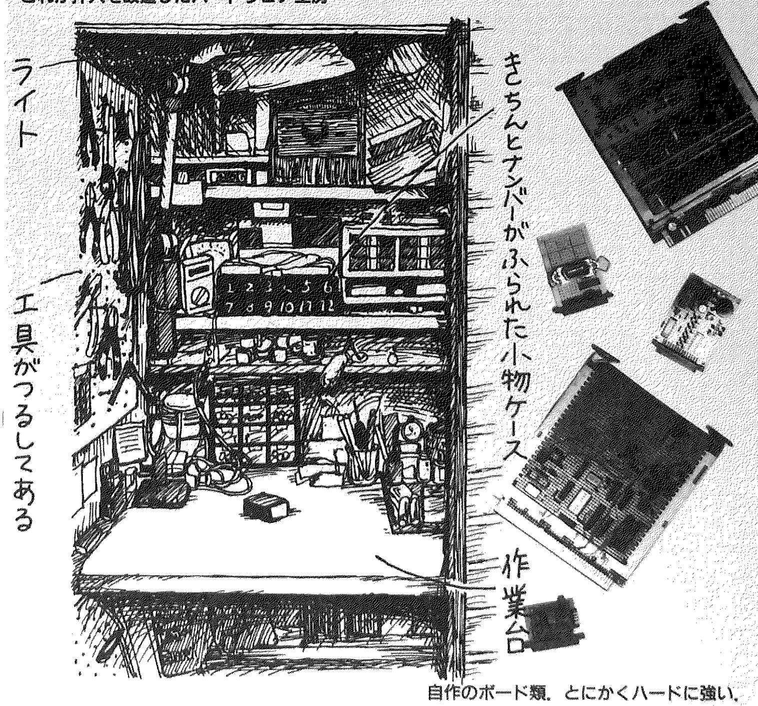
日経MIX PC9800会議の議長を務める小高氏にとって必需品であるモデムは，OMRONの2400bpsのものを使っている.

小高氏は，今まで購入したマシンを売却せずにすべて残してある．これは，PC-9801の世代ごとにソフトウェアの動作試験を行うためと，ハードディスクのフォーマットのように時間のかかる作業を予備機で行ったり，故障のときのバックアップのためである．現在は，バックアップ用も含めて，PC-9801本体3台，モデム2台，ハードディスク4台，プリンタ3台，ディスプレイ2台が揃っている．これなら，よほど運が悪くないかぎり，故障が起きても作業を継続することができる.

このほかに，小高氏が開発環境のひとつと考えているものに椅子がある．なんでもパソコン用にOA用の椅子を購入したところ，それまで使用していた椅子ではちょっと長く座っているとひどく疲労していたのが，新しい椅子に換えてからは，少なくとも座っていること自体では疲れなくなったということである．仕事用の椅子は，値段の高いものを使うほうが絶対生産効率が上がる，というのが小高氏の弁である.

TSR（メモリ常駐プログラム）やデバイスドライバをよく書くので，開発はアセンブラを主体に行っている．アセンブラはMASM5.1，エディタはMIFESを使用し，そのほかのツールとしては，ASCIIのMS-

これが押入を改造したハードウェア工房



自作のボード類．とにかくハードに強い.

DOS SOFTWARE TOOLSを1，2，3すべてそろえて，その中からmake, sed, grepなどを常用しているということだ．ただし，sedはプログラミングではあまり使用していない．

また，リンカはMASMに付属しているものか，TASM付属のものを使い，ソースのバージョン管理には，T. Tanaka氏のRCSを利用している．

割り込み関係のデバッグは，既存のデバッガが使いにくいので，割り込みカウンタのようなものを自作して使っている．このように，必要に応じてツールを自作して使うことが多い．

PC-PR406LM用のプリンタドライバは，漢字を詰めて打ち出すものと縦書きで打ち出すものとMSコードのまま打ち出すものの3つを自作して，これらをすべて組み込んだ上で必要に応じて切り替えて使用している．

小高氏は，131MBのハードディスクを4分割して使っている．本当は単一の領域（あるいはなるべく少ない領域）で使用したほうが使いやすいのだが，そうするとひとつの領域のFATのサイズが大きくなってしまって，MS-DOS SOFTWARE TOOLSのgarbageをかけるときに時間がかかったり，信頼性に不安を感じたりするため，やむなく4分割している．

実際にプログラミングを行う際には，まず日本語でどのような仕様にするか書いていって，そのあとはいきなりコーディングしてしまう．

最初にルーチンのおおまかな構造を書き，これをどんどん増やしていって，それから中身のコードを書いていくということである．いままで書いてきたソースを元に，オリジナルライブラリを作成することも考えているが，忙しくてなかなかそこまで手が回らないようである．

プログラミングの情報源 書籍や雑誌，パソコン通信も活用

PC-9801のアセンブラを覚えたのは，技術評論社の『98マシン語』（注1）という本で，この本はいまでもときどき見る．

MS-DOSに関しては，アスキーの『標準MS-DOSハンドブック』（注2），『MS-DOSプログラマーズハンドブック』（注3）のシリーズがとてもよくできているので，完璧に読んで身につけるようにした．アスキーの『MS-DOSエンサイクロペディアVolume. 1，2』（注4）も，内容が充実しているので参考にすることが多い．

また，実際のプログラミングの際に参考にするものとして，アスキーの『PC-9800シリーズテクニカルデータブック』（注5）があるが，新版には誤植が多いので，まだ古い版のものも見ている．

書籍以外の情報源としては，やはりパソコン通信があげられるようだ．なかでも小高氏自身が会議の議長を務める日経MIXと，ASCIIネットPCSから多くの情報を得ている．

また，通信で知り合った人とオフラインで会ったときに詳しい情報交換をすることも多い．

ただし，小高氏自身はわからないことがあっても自分で調べてしまう．結局自分で調べなければほんとうのところはわからないということなのかもしれない．

（注1）『98FAN BOOKS 2 98マシン語　初めて学ぶマシン語の集中レッスン』（藤木文彦著／技術評論社）
PC-9801の8086マシン語を学ぶための入門書．MASMなどの高級なアセンブラを使用するには，MS-DOSについても学ぶ必要があるため，初心者の負担はかなりのものになる．そこで，本書ではあえてBASIC上のアセンブラでマシン語を学習するようになっている．

（注2）『標準MS-DOSハンドブック』（アスキー出版局編著／アスキー）
アスキーの“黒本”として親しまれている本書は，MS-DOSの存在意義から基本的な操作，各コマンドのリファレンス，MS-DOSの内部構造までを解説している．おもにMS-DOS Ver.2.0を中心に解説しているものの，基本的な部分は今でも古さを感じさせない．

（注3）『MS-DOSプログラマーズハンドブック』（アスキー書籍編集部編著／アスキー）
MS-DOSの開発環境について，具体的な情報を与えてくれる詳細な解説書．MASM，LINK，LIBなどのマシン語開発ツールの機能を詳しく説明するとともに，MS-DOSの特徴的な機能の利用方法，エラー処理ルーチンの作成など多くの実例を示して解説している．

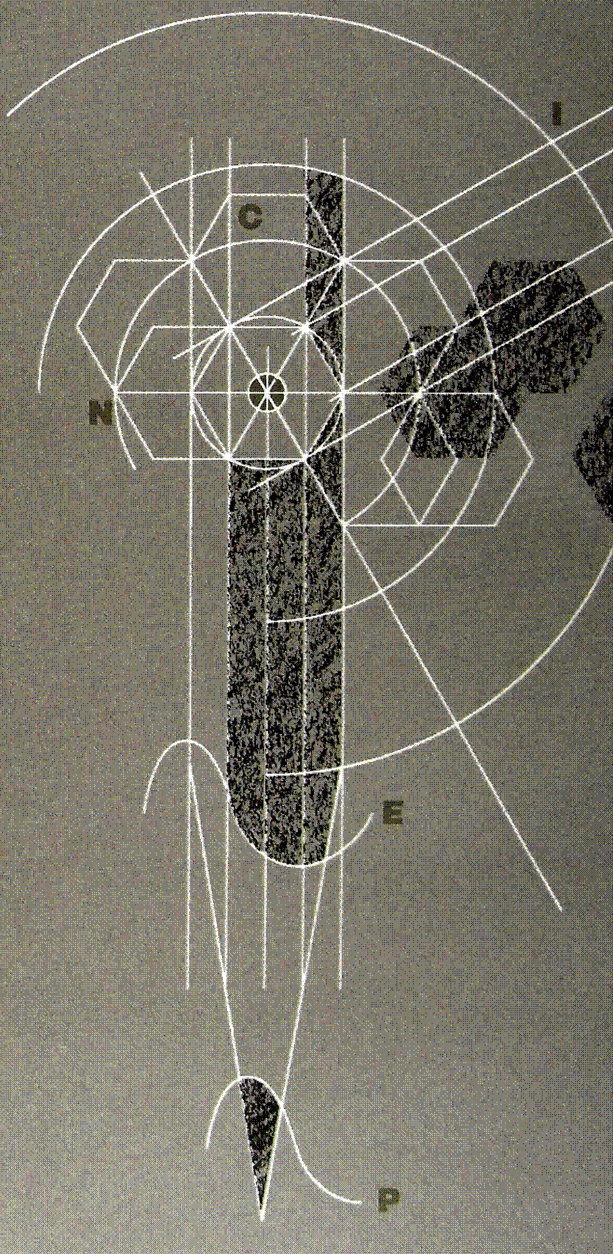
（注4）『MS-DOSエンサイクロペディアVolume1，2』（マイクロソフトプレス編/アスキー）
米国Microsoft Pressから出版されている「The MS-DOS Encyclopedia」の翻訳書であり，MS-DOSのもっとも包括的な解説書である．プログラマを対象としているため，詳細な技術解説およびプログラミングに関する具体的な事例がとりあげられている．

（注5）『新版PC-9800シリーズテクニカルデータブック』（アスキー出版局テクライト編／アスキー）
NEC PC-9801シリーズ全機種を対象にしたデータブック．ROM BIOSのパラメータやスロットバスのタイミングなど，ハード・ソフト両面にわたって詳細なデータを掲載している．1990年に，それまでに出ていた増補版をまとめた総集編が出版された．

祐安重夫

プログラマの道具学

第1回 鉛筆



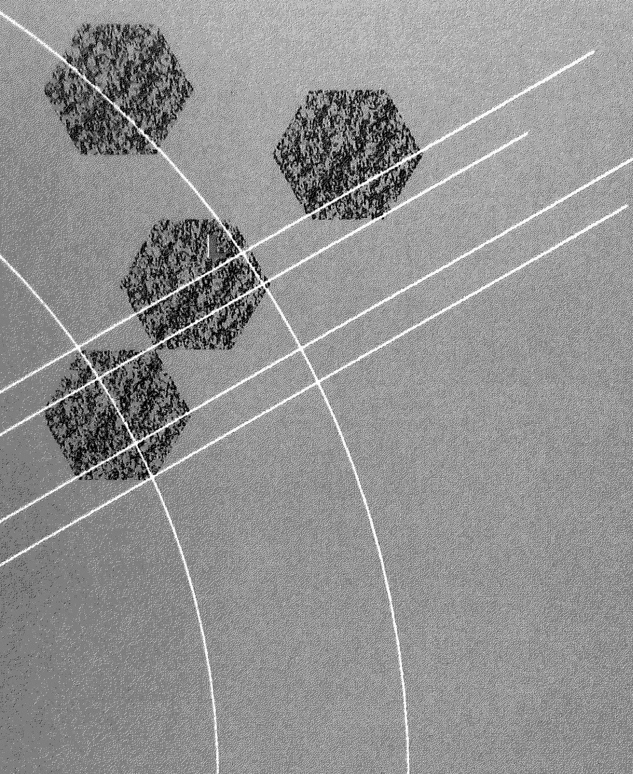
かつては，ソフトウェア・ハウスは紙と鉛筆さえあればできるといったことが，よくいわれていた．実際に私自身，十年とすこし前になんの設備も資金もないまま，ソフトウェア開発の仕事をはじめてしまったのだった．

大型コンピュータが全盛だったころには，ソフトウェアの開発をおこなう会社がコンピュータを所有することは，価格や規模の面から不可能にちかく，コンピュータのおいてある場所にでかけて仕事をすることはあたりまえだった．大型コンピュータに関しては，それは現在でもあまりかわっていないだろう．

また現在のようにオンラインの開発環境が普及する以前は，プログラムは直接キーボードから入力するものではなく，コーディング・シートとよばれる特別な用紙に，鉛筆でかいていくものだった．それをキーパンチャとよばれる人間がカードにパンチし，そのカードをコンピュータによみこませるという手順をふんで，プログラミングをおこなっていたのである．

ソフトウェア・ハウスが紙と鉛筆さえあればできるといわれたのは，実際にはこういうことだったのだ．

十年ほど前にパーソナル・コンピュータが出現したころは，まだまともなソフトウェア開発環境などなかったので，コーディング・シートが別のかたちで利用されていた．アセンブラ語でプログラミングする場合，普通はコンピュータによってそれを機械語に変換するわけだが，まだアセンブラが用意されていないマシンがおおかったので，それを手で機械語に翻訳すること——ハンド・アセンブルが結構おこなわれていた．その際に変換された機械語を16進数でかきこむための場所を用意した，マイクロプロセッサ用のコーディング・シートなどというものがつくられ，実際につかわれていたのである．



その後の十年間の変化は，かなりおおきなものだった．パーソナル・コンピュータやワークステーションが普及し，ソフトウェア開発の現場でもひとりに1台以上のコンピュータやターミナルが用意されるのは，ごくあたりまえのことになってきた．また高機能のエディタやデバッガといった，オンラインでのソフトウェア開発を支援する道具――ソフトウェア・ツールが用意されるようになったため，コーディング・シートにプログラムをかくといった作業は，完全に過去のものになってしまった．

先進的なソフトウェア開発の現場では，システムの設計からコーディング，各種のドキュメントの作製といった作業まで，そのほとんどをコンピュータによるオンラインの環境でこなすことができるようになってきている．私自身も仕事のほとんどを，ワークステーションにむかってキーボードやマウスを操作しながらおこなっているし，手のとどく範囲には5台のコンピュータのディスプレイがある．もちろんこの原稿も，コンピュータにむかって，キーボードから入力していることはいうまでもない．

念のためにつけくわえておくが，先進的なソフトウェア開発環境とは，たとえばΣプロジェクトがめざしてきたものとは，まったくちがう．個人的には，このようなプロジェクトが存在したことさえわすれてしまえる日が，できるだけはやくくることをいのっている．

もっとも鉛筆に完全にわかれをつげることは，とうていできるわけはない．なるべくやらないようにしてはいるのだが，大型コンピュータをつくっているメーカなどから仕事をひきうけると，手でかかなければいけない書式の用紙をあてがわれて，それをつかった文書の提出をもとめられることがある．それだけでも不愉快なのに，その書式のもとになっているかんがえかたが，今や過去の遺物になっているような，崩壊したソフトウェア工学上の技法であったりすると，かなり腹がたってくる．もっともこういったメーカが，じつはΣプロジェクトの参加企業でもあることをかんがえると，なかなか興味ぶかくはあるが．

それは別にしても，日常のメモのようなものまですべてキーボードから入力してプリントするなどというのは，あまり現実的ではないので，鉛筆やボールペンをすてるわけにはいかないのは当然である．現在は鉛筆ではなく，おもに0.9mmの芯のシャープペンシルをつかっている．鉛筆やボールペンはペンのように先がしなわないため，腱鞘炎になりやすいそうだが，このシャープペンシルはばねで芯がひっこむようになっているので，気のせいかもしれないが手がつかれにくいようだ．また2Bの芯をつかっているので，やわらかい力ですむ．

Macintosh

Macintoshのプログラミング環境 1

Macintoshプログラミングとはどのようなものだろうか

大座畑 重光　マッキンテリジェンス(株)

1 Macintoshとは

Apple Iが世に出たのは1976年のことだ．それはパーソナルコンピュータの誕生であり，記念すべき年となった．開発者は，後にApple社を創立するスティーブ・ウォズニアックとスティーブ・ジョブズである．

1977年には，AppleIIが発売され，これもすごく売れた．AppleIIには，ビデオRAMがついており，多くのTVゲームもあった．何万というアプリケーションソフトが開発され，アメリカの学校教育現場でも数多く利用されている．また，拡張スロットがいくつかついていて，オープンアーキテクチャであるということも開発者には非常に魅力的な要素であったのだろう．発売後3年たった1980年9月までに約13万台を販売した．1980年夏には，NCCにおいてAppleIIIが発表されたが，これは結果的にあまり売れなかった．

1983年のApple社株式総会では，Lisaコンピュータが発表された．Lisaという名称は，Local integrated software architectureの略との説と，ジョブズの子供の名前からとったとの説もある(＊1, 4)．Lisaは，Xerox社のPARC（パロアルト研究センター）で長年研究されてきたAltoコンピュータから種々の技術を受け継いでいる．ビットマップディスプレイとマウスがつき，アイコンと呼ばれる絵文字によって，エンドユーザーに，よりよい操作環境を提供しようとした最新型のものであった．オペレーティングシステムは，シングルユーザー，マルチタスクであり，アプリケーションとしてLisa Office Systemと呼ばれた7本の統合ソフト(LisaCalc, LisaProject, LisaGraph, LisaList, Lisa Draw, LisaWrite, LisaTerminal)が提供されていた．当時，これらは非常にすばらしいものであった．

1984年1月には，Macintosh(愛称Mac）が発表され，いよいよMacintoshの時代が到来した．Macintoshでいちばんユニークなところは，ROMの中身のソフトウェアである．これは，ウィンドウマネージャ，イベントマネージャ，メニューマネージャ，リソースマネージャなどに整然と類別化され，それらを構成するルーチンは500以上ある．そのルーチンひとつひとつについて詳しく解説した『Inside Macintosh』という書籍がAddison-Wesley社から出版されている（＊3)．“Macintoshのバイブル”とでもいうべきものだ．この本は，ユーザーインターフェースのガイドラインについても規定している．プログラマがこれを守ってプログラミングする限りは，アプリケーションの一貫性が保たれる．この一貫性こそが，エンドユーザーが

異なるアプリケーションを使う場合に，“使う意欲”をわかせるひとつの要因になっている．

エンドユーザーにとって，どのようなアプリケーションに対しても，似ている環境，見慣れている環境，一貫している環境が与えられることは，はじめて使うアプリケーションであっても，使う意欲がわいてくるものである．これは，コンピュータに不慣れなユーザーはもとより，普通のユーザーにとってもありがたく重要な点である．

このような一貫性のあるアプリケーションは，従来のプログラミングスタイルでは実現できなかった．ユーザーインターフェースのガイドラインやToolbox（ROM）が与えられていない開発環境では，プログラマは好き勝手にユーザーインターフェースを構築してしまうからだ．

前述したように，Toolboxは非常によく構成された500以上のルーチンからなる．プログラマは，これらのルーチンを使いこなすためのドキュメンテーション『Inside Macintosh』（かなりの大著）との格闘を強いられる．この格闘に打ち勝ったプログラマのみが，Macintoshのすぐれたプログラムを開発できる可能性が与えられる．いわば，Macintoshのプログラミング作法というものをマスターしなければならないわけだ．しかし，いったんマスターしてしまえば生産性と信頼性は飛躍的に向上する．

上述のように，ユーザーフレンドリな一貫性をもつユーザーインターフェースの実現は，プログラマの（Inside Macintoshを熟知しなければならないという）苦労の上に構築されるものである．

このように，Macintosh上のプログラミングはなかなか容易ではなかった．エンドユーザーとプログラマとの間には，かなりの隔たりがあった．エンドユーザーは，アプリケーションのユーザーインターフェース部分をリソースエディタを使って，自分に合うようにグラフィカルに容易に修正できるが，与えられたアプリケーション本体を自らの要求で修正・拡張することはほとんど不可能に近い．エンドユーザーからプログラマへのスムースな移行を成し遂げられるようなソフトウェア構築ツールの必要性は誰もが感じていた．これを実現するには，テキストとグラフィックスの統合化だけでは不可能で，さらに第3の要素，インタラクション（相互作用）を統合化する必要があった．そこに登場したのがHyperCardだったのである．

2 HyperCardとは

HyperCardによって，Macintoshのプログラミングも“取りつきやすさ”が実現された．ある範囲のプログラムは誰にでも容易に構築できるようになり，HyperCardの範囲外の不足な機能だけは，従来のToolboxを使ってプログラミングし，それを組み込め(XCMD)ば，要求を満たすプログラムを，より容易に作成できる．

HyperCardは，すべての人が情報を記憶・加工・検索し，自在にやりとりできるツールとして，Macintoshに新しいパワーを吹き込むことに成功した．HyperCardは，Macintoshの誕生後3年半を経過した1987年8月にボストンで行われたMacWorld Expo '87（Macintoshだけを対象としたショー）でデビューした．そこに“スタックウェア”という新しい概念が登場し，ソフトウェアの世界に大きな爆弾を投じることになった．HyperCardの最大の功績は，コンピュータに不慣れな一般大衆にもそのパワーを解放した点にある．さらに，これはソフトウェアにおけるひとつの革命で，ソフトウェアにまつわる多くの障壁を打ち砕くパワーを秘め，ソフトウェア危機を緩和，あるいは乗り越える可能性をもっているといえる．

HyperCardは，テキスト，グラフィックス，ビデオディスク，人の声，音楽，アニメーションなどの情報を組み合わせ，あたかも人が連想するのと同じようにすばやく活用できるツールキットといえる．これは，QuickDraw（ROM内ルーチンであり，その詳細は『Inside Macintosh』に掲載してある）やMacPaintの作者として知られるビル・アトキンソンの作である．現在，HyperCardはMacintoshに標準バンドルされ，初心者から上級プログラマまできわめて幅広い層に受け入れられ愛用されている．

―――大部分のプログラミング言語はデータをもっていない．処理したいデータは，すべて定義するかタイプインしなければならない．HyperCardには数千の単語，カード，絵がはじめから与えられている．しかも，それらを見ることができる．すべてのデータを見ることができるので，プログラミングがはるかに容易になるのだ．（＊5）
Ted Kaehler, Apple Computer, inc.

3 Macintoshとほかのパソコンのちがい

Macintoshがほかのパソコンと大きく異なるところは，Macintosh上のすべてのアプリケーションのユーザーインターフェースが統一されるように，プログラマや開発者に対して，そのガイドラインを示していることだ．さらに，プログラマがそのガイドラインを自然に実現しやすいように

「Toolbox」という名で呼んでいる500以上の完成されたルーチン類がROMの中に提供されていることだ．これは，「Macintoshの心臓部」と呼んでもいいだろう．Toolboxを書いたのは，アンディ・ハーツフェルドやビル・アトキンソンをはじめとする5人のメンバーである（＊4)．たった5人でこれだけの仕事を成し得たことは偉大であるとしかいいようがない．

このROMの中に約20KBを占めるQuickDraw（ビル・アトキンソンの名作）があり，これはMacintoshのグラフィックスの処理部分をつかさどるきわめて重要な描画ルーチンの集まりである．この描画ルーチンは，ユーザーの動きにすばやく対応するスピードを実現している．

また，「リソース」という概念がある．これはApple社によるすばらしい発明のひとつに数えられる．ここでは，その利点などを述べないが，画面に表示されるユーザーインターフェースの部分のいろいろなオブジェクト，たとえばメニュー，文字列，ウィンドウ，アイコンなど，さらにコードもリソースである．これにより，ソースプログラムがなくてもリソースエディタを使ってエンドユーザーがある程度自分に合うようにユーザーインターフェースを変更することが可能である．

4 Macintoshのソフトウェア開発環境

Apple社が提供するMacintoshの開発環境としては，MPW（Macintosh Programmer's Workshop)，MPW Object Pascal，MPW C，MPW C++，MPW Assembler，それにMacAppがある．MPW Object Pascalは，オブジェクト指向プログラミングを行うためのObject Pascalをサポートしている．Object Pascalは，Pascal言語の開発者として知られるスイス連邦工科大学（ETH）のNiklaus Wirth教授をApple社に助言者として招いて開発されたものである．これは，Lisaで可能だったClascalというPascalのオブジェクト指向バージョンから引き継がれた．

また，MacAppは，Lisaのアプリケーションを開発するためのアプリケーションフレームワークであるLisa Toolkit（これはClascalで書かれた）から引き継がれたオブジェクト指向プログラミングライブラリであり，多くのMacintoshアプリケーションプログラムに共通の特徴を実現している．また，メニュー，デスクアクセサリ，ウィンドウのスクローリングやリサイジング，プリンティングのような標準のユーザーインターフェースの特性を自動的に操作し，さらに強力な高級デバッガを含んでいる．MacAppはObject PascalやMPWアセンブリ言語のオブジェクト指向マクロで書かれている．Object Pascalの利点については『TURING MACHINE Vol. 3, No. 4』（＊9）を参照して欲しい．

5 Appleのアプローチ

パーソナルコンピュータは誰にでも容易に操作でき，人間の行ういろんな情報の処理を手助けしてくれることが理想だが，残念ながら現状ではそういうふうにはできていない．これを実現できるようになるまでには，多くの，とくにソフトウェアの問題を克服していかなければならないであろう．人とマシンの知的インターフェースの実現には，人間の常識や専門知識などが形式化されて，知識ベース（Knowledge Base）としてあらかじめマシンに格納されていることが必要だ．それによって，システムが人の個性に合わせて応答してくれる可能性がでてくる．それには，AI的なアプローチが必要になってくる．

Apple社のスカリー会長は，Appleの将来の展望について次のように述べている（＊2, 6)．

①グラフィックス
②ヒューマンインターフェース
③コミュニケーション

などの技術力を強化し，「グラフィックスとコミュニケーションの機能を強化して使いやすくしたパーソナルコンピュータ」すなわち，第2世代パーソナルコンピュータの開発を推進していく．

これからのパーソナルコンピュータは，ハードウェアでなくソフトウェアがキーになる．そのために，Appleのアプローチとして，

①どうしたら創造的な道具を人に与えることができるか
②どうしたら複雑に入り組んでいる情報をわかりやすく人に伝えることができるか
③あるいは，そうした情報をいかに容易に扱うことができるか
④そういう道具をどうやって作っていくか

ということを重視していきたい（＊4, 6)．

このへんの内容は現在でも普遍的である．アラン・ケイ博士たちが取り組んでいるビバリウムプロジェクト（＊7）の成果も1995年頃には出てきそうである．また，最近Apple社からスピンアウトしたビル・アトキンソンは今後も名誉アップルフェローであり，ゼネラル・マジック社の会長としてPer-

sonal Intelligent Communicator (PIC) の開発に取り組んでいる（*10,11）．これは，パーソナルコンピュータというよりは，電話機のようなものに近く，机上だけでなく，より広範囲に実生活で役にたつものだそうである．ゼネラル・マジック社には，Apple社のスカリー会長も取締役として参加している．コンピューティングの世界も徐々に拡大され，多くの人々が参加できるようになってきたことは非常におもしろい．

*1 マイケル・モーリッツ著（青木訳）『アメリカン・ドリーム』，1985，二見書房
*2 『日経パソコン』10-20，pp.152，／1986．
*3 Apple Computer,Inc.『Inside Macintosh』，Addison-Wesley Publishing Company
*4 大座畑「アップルから始まったパーソナルコンピューティング」，THE COMPUTER，pp.84-91，日本ソフトバンク（1988.2）
*5 C．ケーラー著（大座畑訳）『HyperCard―その偉大なパワー―』，アジソンウエスレイ・トッパン情報科学シリーズ，トッパン（1989,10）
*6 大座畑「ヒューマン・コンピューティング環境の未来」，MAC+CYBER，Vol.8,No.18，pp.42-49，佐久書房（1989.12）
*7 大座畑「アラン・ケイの講演」，bit Vol.22,No2，pp.40-43，共立出版（1990,2）
*8 大座畑「ぼくらの時代のNEW COMPUTING STYLE―コンピュータが友達になる日」，THEソフトバンク'90，pp.16-19，日本ソフトバンク（1990,3）
*9 大座畑「MPW Object Pascalのオブジェクト指向について」，TURING MACHINE，Vol.3，No.4，pp.129-136，スペック（1990.8）
*10 坪田「NCインタビュー」，日経コンピュータ，pp.56-59，日経BP社（1990.10.22）
*11 大座畑「ビル・アトキンソンの旅」，bit，Vol.23,No.1，共立出版（1991.1）（予定）

アプリケーション

Toolbox (ROM部分)
Resource Manager
QuickDraw
Font Manager
Toolbox Event Manager
Window Manager
Control Manager
Menu Manager
TextEbit
Dialog Manager
Desk Manager
Scrap Manager
Toolbox Utilities
Package Manager

ハイレベルソフトウェア (RAM部分)
Binary-Decimal Conversion Package
International Utilities Package
Standard File Package

オペレーティングシステム (ROM部分)
Memory Manager
Segment Loader
Operating System Event Manager
File Manager
Device Manager
Disk Driver
Sound Driver
ROM Serial Driver
Vertical Retrace Manager
System Error Handler
Operating System Utilities

ローレベルソフトウェア (RAM部分)
RAM Serial Driver
Printing Manager
Printer Driver
Apple Talk Manager
Disk Initialization Package
Floating-Point Arithmetic Package
Transcendental Functions Package

Macintosh ハードウェア

▲Macintoshのシステム概念

COLUMN

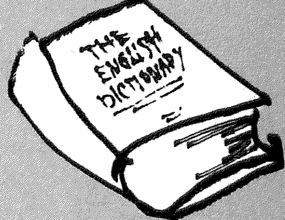
MacWorld Expo/Tokyo

Apple社は，Macintosh誕生後，毎年2回以上「MacWorld Expo」というMacintosh専門の展示会を開催している．1991年2月には，はじめて東京で開催される予定だ．Apple社のジョン・スカリー会長，アップルフェローのアラン・ケイ博士など多くの講師が予定されている．

現在，Macintoshはシリーズ化されて商品展開を行っている．つい最近（1990年10月）もニューモデルが3機種が追加された．シリーズ全体を通していちばん低価格のMacintosh Classic（198,000円），カラーが使える機種としてはもっとも低価格のMacintosh LC，それにIIシリーズでもっとも低価格のMacintosh IIsiである．画期的な価格設定によって，さらに飛躍的な市場の拡大が期待できるであろう．

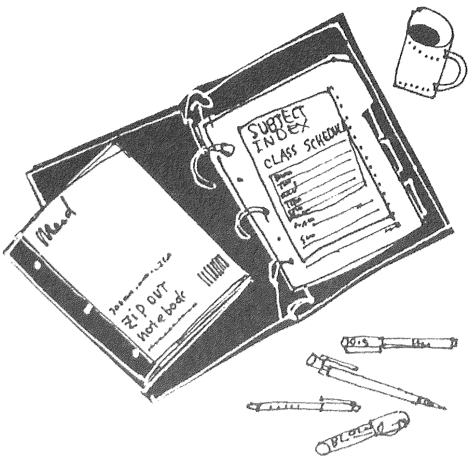
実践C塾 第1回

MS-DOS上の C処理系の現状



塾頭 水居 浄

この連載では，MS-DOS上の代表的なC言語処理系（おもにMicrosoft CとTurbo C）について，いろいろと技術的な話をしていきたいと考えています．各処理系の特徴や，MS-DOS上で実際にプログラミングする際のさまざまな問題点，およびそれらの解決方法などについてまとめた資料的価値のあるものをめざしていく予定です．なるべくていねいに説明するつもりですが，やはりMS-DOSとC言語についてある程度のプログラミング知識があることを前提とさせていただきます．



Cという言語は，意外と奥の深い言語です．ある程度ほかの高級言語の経験があれば，一見簡単に見えますし，また実際すぐにプログラムが書けるようになります．しかし，プログラムが書けるということと，言語そのものをきちんと理解しているというのはまた別問題なのです．私自身，もう5年以上の間いろいろな立場でCと付き合ってきていますが，いまだに新しい発見があり，そのたびに奥の深さを実感しています．現実には，そういった細かい知識はなくてもプログラムは書けますし，実用上とくに役にたつ知識ではないことも多いのですが，そういう発見など，いろいろな情報を記事にしていき，少しでも世の中のプログラマの方々の役にたてれば幸いだと考えています．

Illustration:MASAMI NOZUE



1 Cの一般的な特徴

Cの誕生
UNIXを記述するために生まれたシステム記述言語

C言語は，よく知られているように，AT&Tベル研究所でDennis M. Ritchieにより開発され，1972年に誕生したプログラミング言語です．手続き型言語の一種であり，BCPLやBといった言語から進化してきたものであるという話は，すでにいろいろな文献で語られていますので，ここで改めて繰り返す必要はないと思います．

Cが開発された目的は，当時アセンブラで記述されていたミニコン用OSであるUNIXを高級言語で記述しなおすことにより，記述性と保守性を向上させる（機能改善を容易にし，バグを取りやすくする）ことでした．Cは一応高級言語ですから，ある程度抽象化した計算式や制御構造の記述が容易であるという，高級言語の特徴を備えています．その一方で，Cはポインタや多くの演算子をもち，アセンブラレベルに近い細かい操作の記述が可能ですから，かなりの点でアセンブラの代用品として使用できます．こういった特徴が，それまでアセンブラのみで記述されていたOSを高級言語で記述しなおすことを可能にしたのです．

このように，Cはミニコン上でOSなどを記述するための，いわゆる「システム記述言語」として誕生したものなのです．

Cの特徴
常識のウソ，ホント？

Cの特徴としては，一般に次のような点があげられています．

- アセンブラレベルの細かい記述が可能
- コンパクトで効率のよいコードを生成する
- ポータビリティがよい　(?)

ここで，それぞれの特徴について述べてみます．

アセンブラレベルの細かい記述が可能

Cには数多くの演算子があり，しかもそのいくつかは，一般的なCPUの機械語命令に1対1で対応します．たとえば，ビットごとのデータ操作の演算子を8086CPUの命令に対応させると，表1のようになります．

また，Cでは扱うデータのメモリ上でのイメージをほとんど直接的にプログラマに見せています．たとえば，文字型データはその文字に対応するコード値（通常はASCIIコード）で表され，そのまま整数値として演算することも可能です．また，文字列データは文字型の配列としての実体しかもたず，他の言語のように抽象化された構造をもっていません．さらに，それぞれのデータのアドレスを取得することが可能で，そのアドレス値はメモリアドレスに直接対応します．したがって，ポインタ変数を使ってメモリ空間上の任意のアドレスに直接アクセスすることが可能です．こういった特徴が，Cをアセンブラの代用品として実用的なものにしているのです．

コンパクトで効率のよいコードを生成する

Cの生成するコードは，一般的に他の言語のコンパイラが生成するコードよりコンパクトなものになります．これは，Cがデータやプログラムフローをあまり抽象化せずにダイレクトにユーザーに見せていることや，オーバーフローなどの演算エラーのチェックをほとんど行わず，ユーザーの責任に任せているためです．したがって，Cでプログラムする場合は，他の言語では処理系側が自動的にやってくれるようなエラーチェックまで，ほとんどすべてをユーザーのプログラムで面倒を見る必要があります．

すなわち，Cで書いたプログラムが速いのは，一種の手抜きをやっていることによる部分が大きいので，現実には，エラーチェックなどをていねいに処理していくと，必ずしも他の言語より速くはならないことも多いのです．また，そういった面倒を見る部分をきちんと記述できないような初心者がCを使うと，往々にしてとんでもない失敗に陥ることがありますから，注意が

表1：演算子と8086CPUの命令の対応

演算子	8086の命令
~	NOT
&	AND
|	OR
^	XOR
<<	SHL
>>	SHR，SAR

必要です．こういう点でも，Cはアセンブラに近い特異な高級言語であるといえます．

ポータビリティがよい(?)

Cで記述されたプログラムは，ポータビリティがよいと一般にいわれています．しかし，この言葉には注意が必要です．とくにパソコン上では，現実問題としてこの言葉はウソになるからです．C言語に関するANSI規格（後述）が成立したのはつい最近のことです．それまで，Cの言語仕様にはランタイムライブラリ関数の仕様は含まれていませんでした．したがって，使用する処理系によってライブラリ関数の仕様は細かい点で食い違っていることが多いのです．

また，Cはその誕生の経緯から，UNIX環境で実行されることを想定しています．UNIXが今のようにワークステーションのOSとして使われ，細分化されるようになる前は，UNIX環境というものはどのマシンであろうと似たようなものであり，Cで書かれたプログラムも必然的にいろいろなマシンでほとんど修正せずに，再コンパイルするだけで動作できるようになっていました．また，Portable C（後述）というコンパイラそのものの移植性がよい処理系が標準的に存在していたため，たいていのUNIX環境ではこのPortable CをもとにしたCコンパイラが使われていました．こういった歴史が，「Cはポータビリティがよい」という言葉をつくってきたのです．

もちろん，現在はUNIXといってもさまざまなベンダーが固有のウィンドウシステムなどの機能を拡張して販売していますから，そういった拡張部分に依存するプログラムは当然，移植性の低いものになります．しかし，コンパイラそのものはPortable Cの子孫がほとんどであり，言語仕様や標準ライブラリ関数の仕様などの違いに悩まされることは，比較的少ないものでした．

一方，パソコンの世界では，同じMS-DOS上で動作するといってもさまざまなC処理系が存在し，それぞれ細かい仕様が異なっています．もちろん，最近のパソコン上のコンパイラはANSI規格の仕様を高いレベルで満たしているものが多く，ANSI規格の範囲内ではかなりの互換性があります．しかし，MS-DOS上では現実に実用的なプログラムを記述しようとすると，処理系に固有のライブラリ関数や拡張言語仕様を使わざるを得ない場合が多くあるのです．また，MS-DOSはOSとしては非常に貧弱なものですから，高度なアプリケーションを記述するには標準的なシステムコールだけでは機能が不足します．このため，著名なアプリケーションの大半はVRAMなどのハードウェアを直接操作するようなプログラムになっています．したがって，MS-DOS上のプログラムは，あまり移植性のよくないものになってしまうことが多いのです．

C処理系の歴史
Ritchieのコンパイラからパソコンへ

最初に誕生したCは，前述の通りAT&Tベル研究所のRitchieによって開発されたものです．このコンパイラはDEC社のミニコンPDP-11用で，UNIX Version 5はその90%をこのコンパイラで記述されたといわれています．このRitchie仕様のコンパイラは，その後いくつかの別なマシンにも移植されましたが，広くさまざまなマシンやOS環境へ広まるには至っていません．そういう意味で重要な役割を果したのは，後に同じベル研究所のSteve Johnsonによって開発されたPortable C Compilerと呼ばれるコンパイラでした．このコンパイラは，コンパイラ自身の大部分がCで記述されているとともに，マシンのアーキテクチャに依存する部分が整理されており，比較的少ない労力で他のマシンに移植できるという特徴をもっていました．このPortable Cの登場によって，CとUNIXはさまざまなマシンへ移植されていき，今日の隆盛を見るに至ったのです．現在でも，多くのUNIXマシンが標準搭載しているCコンパイラは，このPortable Cの子孫が大半を占めています．

1970年代末にパソコンが登場すると，このパソコン上でCを使うことはできないか，と考える人々が少なからず現れました．実際には，パソコン上でCが使われるようになったのは1982年ころからです．最初は，8ビットのCP/Mマシンなどにサブセット化されたCがインプリメントされました．代表的なのがBrain Damaged Software社から発売されたBDS Cなどです．しかし，8ビットマシンではメモリ空間が狭いことや，当時は外部記憶装置も容量が小さく低速であったことなどハードウェア環境が貧弱であったため，フルセットのCコンパイラを実装するのはなかなか困難でした．それに，いかにCがコンパクトなオブジェクトコードを生成するとはいっても，人間がアセンブラで記述した

ワンラインNEWS
ソフトハウスが新作ソフトの商品名に頭を抱えている．日本では先願登録で先に名称だけを商標登録できるため，商品名は出つくしているといわれる．

ものにはおよびません．結局8ビットパソコンの時代には，Cは相対的にはまだまだ大きな言語処理系であり，開発言語の主流とはなりえませんでした．Cが広く使われるようになったのは，16ビットパソコンの登場以降です．

1981年にIBM PCが登場し，パソコンの主流は16ビットマシンへと移行していきました．ハードウェアの進歩には凄まじいものがあり，1980年代中頃になって，パソコンが実用的な存在となるにつれて，Cの存在意義は多少変わってきました．それまでアセンブラとBASICが主流だったパソコンのプログラミングに，Cが多く使われるようになったのです．これは，Cが前述のようにアセンブラの代用品としての実用性を備えていたことや，パソコンがある程度高速化・大容量化してフルセットのCコンパイラ処理系が容易に動作できるようになってきたこと，さらにはアプリケーションソフトウェアに対するユーザーの要求が高度になり，プログラムが巨大化してアセンブラでは開発・保守が困難になってきたことなどが理由です．

現在では，パソコンのソフトウェア開発はCが圧倒的な多数派となっており，是非はともかく，まったくプログラミング経験のない初心者でもいきなりCからはじめるというケースも増えています．C関連の書籍も多数出版され，「プログラミング言語C」(B. W. カーニハン，D. M. リッチー著／共立出版株式会社）はこの分野の書籍としては記録的なベストセラーとなるに至っています．

ANSI規格
これでようやく一人前

ANSI(米国標準規格協会)は，日本におけるJISに相当する機関で，さまざまなプログラム言語の規格も制定しています．1980年代に入って，AT&T以外の会社によって開発されたC処理系が多数出回るようになってくると，Cについても，他のFORTRANなどの高級言語と同様にANSI規格化されるのを望む声が大きくなってきました．これはCの言語仕様の細部が曖昧であったために，さまざまな処理系の間での互換性が著しく低くなっていたからです．

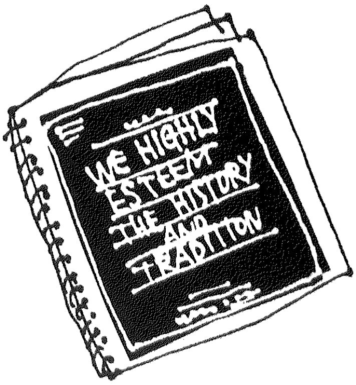
ANSI規格の骨格が固まる前は，Cに関する標準仕様は，俗に著者の名前から「K&R」と略称される「The C Programming Language」(邦訳「プログラミング言語C」)という書籍に記されている仕様であるといわれていました．しかし，この書籍はC言語の学習用のテキストではあっても，詳細な仕様書ではなく，細かい点は実際にRitchieのコンパイラを触ってみないとわからないという状態でした．実際，当時はCの言語仕様の標準はこの書籍ではなく，Ritchieのコンパイラそのものであったというのが正しいのです．もちろん，こういう状況は実用的な言語処理系の仕様としては好ましいものではありません．きちんと明文化された詳細な仕様書が欲しいという要求もまたANSI規格化を促進しました．

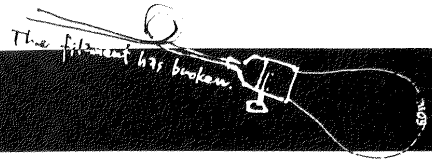
ANSI規格化の作業は，1983年に開始され，最終的に仕様として確定したのは，1989年末のことです．これほど長い時間がかかったのは，ちょうど国際的にCの人気が高まった時期と重なったことから，実にさまざまな質問や要求が数多く委員会に寄せられたためです．このANSI規格の大きな特徴としては，

- 言語仕様を細部にわたって詳細に規定し，規格と処理系依存部分とを明確に区分した．
- それまで言語仕様には含まれないとされてきた，標準ライブラリや環境の仕様を言語仕様の一部として規定した．
- 関数のプロトタイプ宣言や，const，volatile データ修飾子などの新しい機能を導入した．
- 非UNIX環境や，非ASCII系文字コードへの対応をある程度行った．

といったことがあげられますが，基本的にはそれまで使われてきた一般的なUNIX上のC処理系との互換性はできるだけ維持するという方向でまとめられました．これは，10年以上におよぶCの発展過程において蓄積されたソフトウェア資産が，すでに無視できない存在になっていたからです．

規格が最終確定してからまだあまり時間が経っていませんから，現在のところ，完全にANSI規格を満たしているC処理系はほとんどありません．

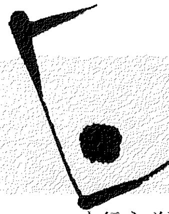




2 MS-DOS上のC言語

前述のように，パソコン上でCが幅広く使われるようになったのは，16ビットパソコンとMS-DOSが普及してからのことです．1990年代を迎えた現在，32ビットパソコンも身近なものになってきましたが，いまだにパソコン用OSの主流はMS-DOSです．そこで，MS-DOSとはどのような特徴をもつ環境であるか，およびMS-DOS上のC処理系にはどのような特徴があるかをここで整理しておきたいと思います．

8086系CPU
ああ！　セグメント



MS-DOSはIntel社の開発した8086系のCPUで動作するOSです．8086系CPUでも，80286以降のCPUではプロテクトモードという動作モードが追加され，アドレス空間が拡大されています．しかし，MS-DOSが動作するのは，基本的に相変らず8086互換のリアルモードです．したがって，アドレス空間は最大1MBまでしか使用できず，しかも16ビットのセグメント値と16ビットのオフセット値を組み合わせて任意のアドレスにアクセスします．このようなセグメント化されたアーキテクチャは，本来Cが想定している環境ではありません．このため，アドレスサイズが複数存在するという，他のCPUにはない問題が発生します．

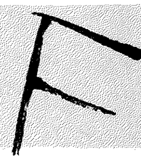
メモリアドレスという概念は，Cでは非常に重要です．ポインタ変数は直接このメモリアドレスの値を保持するからです．このため，8086用のCでは，ポインタをセグメント値とオフセット値の組み合わせ（farアドレス）で表現しなくては，1MBの空間すべてにアクセスすることができません．また，1MBの空間の任意のアドレスに配置した関数をコールするためには，やはりセグメント値とオフセット値を組み合わせたfarコールを行う必要があります．

しかし，プログラムによっては，そんなに大きなコードやデータを必要としない場合も多くあります．そのような場合は，たとえばプログラムの起動時にだけセグメント値を初期化し，あとはアドレスをすべて16ビットのオフセット値のみとして取扱う方法があります．こういう方法を使うと，非常にコンパクトで高速なプログラムが作成できます．むしろ，8086はこういう使い方が中心となることを想定して作られたCPUであるともいえます．つまり，できるだけセグメント値は変更せずに効率よくすませることができるようにしなくてはならないということです．

もちろん，プログラムによっては64KBをこえる大量のデータ領域を使用したり，あるいはプログラムコード自体が64KBに収まらないほど大きくなることもよくあります．この場合は，データをfarアドレスでアクセスしたり，関数をfarコールする必要があります．

こういったさまざまな場合に対処するために，MS-DOS上のC処理系ではメモリモデルや，アドレスの修飾といった概念が導入されました．

メモリモデル
セグメントの有効利用？



メモリモデルという考え方を最初に明確に打ち出したのは，Lattice Cという処理系だといってよいと思います．もちろん，同じくらい古くから使われてきたWhitesmith C86やOptimizing C86などといった処理系でも，今日でいう「スモールモデル」と「ラージモデル」のふたつのモデルを最初からサポートしているものはありました．しかし，これに「ミディアムモデル」と「コンパクトモデル」を追加した4つの基本的なメモリモデル全部を最初にサポートしたのはLattice Cです．

メモリモデルとは，アドレスのサイズによって規定されるものです．つまり，前述のようにアドレスをオフセット値のみ（nearアドレス）で取り扱うか，セグメント値とオフセット値の組み合わせ（farアドレス）で取り扱うかの違いです．コードアドレスとデータアドレスのそれぞれについてこの違いをつけると，表2の4つのメモリモデルが考えられます．

コードアドレスをnearとするモデルは，プログラムコードを64KB以上使用することができません．しかし，関数コールがすべてnearコールで行われるため，一般に実行が高速になります．同様に，データアドレスをnear

とするモデルは，プログラムデータを64KB以上使用することができませんが，そのかわりデータアクセスが高速になります．

最近の処理系では，これら以外にスモールモデルの変型であるタイニィモデルや，ラージモデルの変型であるヒュージモデルをサポートしているものも多くあります．

タイニィモデルとは，コードアドレスとデータアドレスをnearとするだけでなく，さらにコード部とデータ部を同一のセグメントに収めることにより，いわゆる .COM形式のプログラムを作成するものです．.COM形式とすることで，多少プログラムの起動が速くなるといわれていますが，最近のパソコンのCPUや外部記憶装置は高速なので，現実には.EXE形式とほとんど差はありません．また，Cで書いたプログラムの場合，.COM形式にしてもスモールモデルの.EXE形式の場合よりサイズが減少する割合はそれほど大きくありません．つまり，実用上は.COM形式にするメリットはあまりなく，.COM形式でなくてはならない特殊な状況以外ではあまり使われません．もちろん，世の中には.COM形式は美しいといってこだわる方もおられます．

ヒュージモデルとは，ラージモデルの変型ですが，単一のデータ項目からも64KBというサイズの制限を取り除いたものです．前述のように，8086のアーキテクチャでは，セグメント値と16ビットのオフセット値によってデータにアクセスします．したがって，単一のデータが16ビットのサイズ（64KB)に収まらない場合は，セグメント値を変更しながらアクセスするという工夫が必要になります．こういうアクセスを常に行うのは効率的ではありませんから，一般にラージモデルでは単一のデータ項目は64KBをこえないという前提でコードを生成します．そして，単一のデータ項目が64KBをこえることを許すメモリモデルとしてヒュージモデルが導入されたのです．

表２：4つのメモリモデル

メモリモデル	コードアドレス	データアドレス
スモール	near	near
ミディアム	far	near
コンパクト	near	far
ラージ	far	far

このように，データについては64KBという壁はなんとか乗りこえて，大きな項目を扱うことも可能です．しかし，コードについてはこの制限はどうしようもありません．つまり，ヒュージモデルとはあくまでもデータがhuge(巨大という意味)なのであって，プログラムコードについては，単一の関数は64KB以内という制限が残ります．もっとも，64KBをこえるような巨大な単一の関数を記述しなくてはならない必要性というものはほとんどないですし，また関数はデータにくらべて分割が容易ですから，事実上この制限はまったく問題になりません．

ヒュージモデルと同様に，コンパクトモデルで単一データ項目の大きさの制限を解除した，いわゆるヒュージコンパクトモデルといったものも，組み合わせとしては考えられます．しかし，このメモリモデルに対する需要は少ないため，標準的にこれをサポートしている処理系はないようです．

なお，上記の各メモリモデルの呼称は，Microsoft社の定義に後発の各社が従って標準的な呼称となったものですが，他の呼称を使っている処理系もあります．たとえば，Lattice Cではスモール／ミディアム／コンパクト／ラージの各モデルを，それぞれS／P／D／Lモデルと呼んでいます．

アドレスの修飾
より柔軟に

あまりなじみのない言葉ですが，要はnear／far／hugeといった拡張キーワードによってアドレスサイズを指定することです．メモリモデルという概念は，8086のセグメント化アーキテクチャのもつアドレスに関する問題を一応は解決しました．しかし，この方法だけではあまり柔軟性がありません．メモリモデルは，ひとつのプログラム内で扱うコードやデータのアドレスサイズをすべてひとつのサイズに規定してしまうという問題を残してしまったからです．

たとえば，スモールモデルでプログラムを開発しているときに，だんだんプログラムで扱うデータ量が多くなって，64KBのデータセグメントに収まらなくなってしまったといった状況もよくあります．こういう場合，メモリモデルをコンパクトモデルに変更すると解決できるのですが，コンパクトモデルに変更すると，すべてのデータがfarアドレスでアクセスされますから，データのアクセスは一様に遅くなります．ここで，もしも一部のアクセスが遅くてもかまわないデータだけをデータセグメントから追い出すことができたなら，従来通りスモールモデルの効率をほとんど落さずにプログラムを作成できるわけです．同様にプログラム

コードが64KBを少しこえてしまった場合も，一部の関数だけをfarコールするようにすることができれば，効率を落さずにプログラムを記述できます．またMS-DOSでは，後述する事情から画面のVRAMに直接アクセスするようなプログラムも多く書かれるのですが，こういうプログラムも，メモリモデルという方法しかないとコンパクトモデルかラージモデルでないと記述できなくなります．

こういった効率上の要求から，アドレスの修飾という機能が生れました．最初にこの機能をサポートしたのはMicrosoft Cです．ほかにもWhitesmith CなどがMicrosoft Cとは違う文法で類似した機能をサポートしましたが，結局はMicrosoft Cの仕様がMS-DOS上の処理系における事実上の標準になっています．

この仕様では，アドレスの修飾に拡張キーワードを使用します．たとえば，スモールモデルで図1のように記述することができます．

このように拡張キーワードを指定することで，各メモリモデル固有のデータアドレスのサイズと違ったデータを個別に記述することが可能になったのです．したがって，たとえば図2のようにあまり使わないエラーメッセージなどの配列をfar宣言してデータセグメントから追い出すことができます．

また，テキストVRAMに直接アクセスする場合などでも，図3のようにスモールモデルでも簡単に記述できるようになりました．

もちろん，アドレスの修飾はデータだけでなく関数についても行うことができます．たとえば，図4のような記述をすると，func1はnearコールで，func2はfarコールで呼び出されます．

このように，アドレスの修飾機能の導入によって，8086のセグメント化アーキテクチャをさらにきめ細かく活用することが可能になったのです．

MS-DOS
パソコンの標準OS

MS-DOSは8086系CPU上で動作する，シングルタスクのオペレーティングシステム（OS）です．Microsoft社が開発して販売しています．もともとは8080という8ビットCPU用のOSであるCP/Mの影響をかなり受けており，1981年に発表されたMS-DOS Version 1.0の仕様はCP/Mに対する互換性をかなり重視しています．というよりも，むしろこのVer. 1.0は，どちらかというと16ビット版のCP/Mといった感じのもので，機能的に特別に大きな飛躍があるわけではありませんでした．その後，1983年に発表されたVer. 2.0が事実上，最初のMS-DOSらしいMS-DOSといえます．このVer. 2.0では，UNIX風の階層ディレクトリや，インストール可能なデバイスドライバ，バックグラウンド処理の可能なプリントスプーラなどといった新しい機能が導入されています．その後1984年にはネットワークに対応したVer. 3.0，1988年にはEMSメモリをOSレベルでサポートし，ビジュアルシェルを搭載したVer. 4.0が発表されて今日に至っています．

MS-DOSというOSは，基本的にはその名の通り，ディスクファイルの管理に主眼をおいたものです．したがって，画面やキーボード，プリンタなどのそのほかのデバイスについては，OSレベルではあまりたいしたサービスを提供していません．このため，画面制御を行うような本格的なアプリケーシ

図1：拡張キーワード

```
int near  a;     /* データセグメント内に置かれるデータ          */
int far   b;     /* データセグメント外に置かれるデータ          */
int near *c;     /* データセグメント内のデータを指すポインタ    */
int far  *d;     /* 任意のfarアドレスを指すことができるポインタ */
```

図2：配列のfar宣言

```
char far ErrMsg[20][40] = {
        "Syntax Error.",
             .
             .
             .
};
```

図3：VRAMの直接アクセス

```
void ClearScreen()
{
        short far *vram = (short far *)0xA0000000L;  /* VRAM address */
        short i;
        for (i = 0; i < 80*25; ++i)
                vram[i] = 0;
}
```

図4：関数のアドレス修飾

```
int near func1(int a)
{
        printf("near function¥n");
}

int far func2(int a)
{
        printf("far function¥n");
}
```

ョンでは，どうしてもVRAMなどのハードウェアを直接操作したり，機種に依存したBIOSなどのサービスを利用しなくてはなりません．こういった事情から，MS-DOS上の著名なアプリケーションのほとんどは機種依存しています．

各処理系の概要
MS-DOS上のCコンパイラ

最初にMS-DOS上で動作するようになったC処理系が何であるかは，たしかな資料が手元にないため，はっきりとはわかりません．おそらくLattice Cあたりであろうと思われます．ともかく，MS-DOSの誕生からあまり時間をおかずに，その上で動作するC処理系は登場しています．現在は，数十種類のC処理系が販売され，使用されています．以下，おもな処理系について簡単に概要を記しておきます．

Microsoft C
Microsoft社

不動の業界標準

このコンパイラは誕生以来，MS-DOSの世界では業界標準としての地位を維持してきています．MS-DOS自体の開発元であるMicrosoft社が提供する処理系という毛並みのよさと，豊富な機能や，常にベンチマーク性能で上位にランクされる高い性能が評価されてのことでしょう．最初のVersion 1.0が発売されたのは1983年頃で，1990年現在Ver. 6.0が最新版です．しかし，初期のVer. 1.0, Ver. 2.0と，Ver. 3.0以降のバージョンとは実質的には別のコンパイラです．つまり，Ver. 1.0, Ver. 2.0は実はLattice Cの同じバージョンをOEM販売したものであり，自社で開発したものではなかったのです．Ver. 3.0以降は自社開発したコンパイラに切り替えて今日に至っています．

そのMicrosoft C Ver. 3.0は1985年に発売されました．1985年ころは，パソコンにおいてC言語がブームになってきた時期でもあり，MS-DOS上で最も優れた処理系はどれか，といった議論が雑誌を賑わしていました．その当時，人気を二分していたのが，後述するLattice C Ver. 2. X とOptimizing C Ver. 2. Xで，それぞれの支持派による論戦が活発に行われていたものです．しかし，そこへ登場したMicrosoft Cは，こういった論戦を思いがけない方向から決着づけてしまいました．実際，このコンパイラは当時としてはまさに次のような点で画期的でした．

- UNIXライクなコンパイラドライバが標準で添付されている．
- near，farといった拡張キーワードで8086のセグメント化アーキテクチャを有効利用する新しい道を開いた．
- リンカ，シンボリックデバッガを標準添付している．
- 混合言語プログラミングの機能をサポートした．
- 非常に詳細なマニュアルが添付された．

どれも現在ではめずらしくないことになってしまいましたが，当時としては画期的なことでした．そして，Lattice C Ver. 2. Xに対してはコンパイル速度が比較的遅い点，Optimizing C86 Ver. 2. Xに対してはライブラリのソースが公開されていない点ぐらいしか劣る点がなく，画期的な性能と機能で両者を圧倒してしまったのです．

日本では，Lattice Cがライフボート社から販売されていたこともあって，Microsoft C Ver. 1.0, Ver. 2.0はほとんど輸入されていません．Ver. 3.0になって日本版が1986年に発売されています．最初は某A社が248,000円という「神をも恐れぬ値段」で発売しましたが，その後Microsoft社の日本法人であるマイクロソフト株式会社が設立されるとともに，98,000円というなんとか納得できる価格に落ちついています．この日本版もまた次のような点で画期的なものでした．

- コンパイラ自体が日本語対応している．
- 日本語処理ライブラリが追加された．
- マニュアルがきちんと日本語化された．

これも現在ではあたりまえのことですね．しかし，当時のCコンパイラは一部の輸入業者が輸入して販売するも

表3：Microsoft Cのバージョン

バージョン	発売年		おもな特徴
	US版	日本版	
V1.0, 2.0	～1985	—	Lattice CのOEM.
V3.0	1985	1986	自社開発に切り替え．性能と機能の向上．
V4.0	1986	1987	コンパクトモデル，ヒュージモデルをサポート．CodeViewを添付．
V5.0	1987	—	オプチマイザを強化．Editor，Quick Cを添付．
V5.1	1988	1988	OS/2対応．
V6.0	1990	1990(予定)	オプチマイザを強化．PWBを添付．

のであり，日本語処理については「漢字プリプロセッサ」(すでに死語になってしまいました）で対応するだけ，マニュアルについても翻訳したものが付いていればよいほう，といった状態が普通だったのです．

Microsoft C Ver. 3. 0が発売された当時，宣伝文句としては旧バージョン，すなわちLattice Cとの互換性も高いといったことがいわれていました．しかし，実際にはソースレベルである程度の互換性は配慮したものの，内容的にはまったく別のコンパイラであり，あまり互換性が高いわけではありません．なぜわざわざ互換性を捨ててまで自社開発版に切り替えたのでしょうか？　それはMicrosoft社にとってこのコンパイラが戦略的に非常に重要な存在であったからです．1985年までに，Microsoft社は今後同社が販売するすべての製品を基本的にCで開発することを決定していました．そして，そのためには当時のLattice Cでは明らかに力不足でした．そこで，8086系CPUをターゲットに最適化した高性能なオリジナルコンパイラを独自に開発することにしたのです．現在，Microsoft社が販売しているほとんどすべての製品はMicrosoft Cで記述されています．

Microsoft Cは，現在までに表3のようなバージョンが販売されています．

もちろん，それぞれのバージョンにはさらにマイナーチェンジ版がいくつか存在します．このほか，IBMがOEM販売しているIBM C，IBM C/2や，XENIX版，およびUNIX上で動作するクロスコンパイラもあります．なお，価格はVer. 3. 0が395ドルでしたが，少しずつ値上げされVer. 6. 0は495ドルになっています．日本版は（最初のA社版を除いて）Ver. 3. 0～Ver. 5. 1が98,000円と一定でした．

Turbo C
Borland社

元祖速い安い上手い

Turbo PASCALの成功でコンパイラベンダーとしての不動の地位を得たBorland社が，満を持して発売した処理系です．最初のVer. 1. 0の発売は1987年ですから，かなり後発の参入になります．しかし，この処理系は次のような特徴をひっさげて鳴りものいりで登場したため，あっという間にMicrosoft Cと人気を二分するほどの存在になりました．

- 統合開発環境が添付されている．
- コンパイル・リンクが非常に高速．
- 生成コードの質もMicrosoft C Ver. 4. 0に匹敵する．
- 機種依存したグラフィックスライブラリを添付．
- 99. 95ドルと低価格．

言語仕様としては，Microsoft Cに習った拡張キーワードをサポートしており，Microsoft Cとの互換性もかなり高いといった宣伝がなされました．まあ，これはあまり正しくはなかったのですが……（この点はあとで詳細に検討します)．ともかく，低価格にもかかわらず高性能かつ高機能なこのコンパイラは，急速に普及しました．

Microsoft Cは，OEMから自社開発に切り替えて今日に至っていますが，Turbo Cについては，これとは逆ともいえるエピソードが伝わっています．すなわち，Turbo Cの本体はBorland社が完全に自社で開発したわけではなく，別の会社が開発したコンパイラを丸ごと買い取ったものらしいのです．そのコンパイラはWizard Cという名前で1986年ころに販売されていました．もちろん，これには統合環境などないですし，コンパイルも遅いなど，Turbo Cとはかなり違う点が多くあります．結局，このコンパイラを買い取ってから，Borland社内で大改造を行った結果がTurbo Cである，ということのようです．まあ，このエピソードは同社から正式にコメントされたものではなく，あくまで噂話にすぎないのですが……．

発売当初は，Microsoft Cに対してデバッガなど機能面で一歩遅れをとっていたTurbo Cですが，その後，CodeViewをこえたともいわれるTurbo Debuggerを発売し，またコンパイラもバージョンアップを何度か行って高性能化してきています．しかし，コンパイル速度を重視したためか，オプチマイザの改良はあまり行っていません．1990年現在，Turbo CとしてはVer. 2. 0が最新版ですが，これとは別にTurbo C++ Ver. 1. 0が発売されました．Microsoft CがVer. 6. 0ではC++化を見送って性能向上に専念したのと比較すると，戦略的な違いがはっきり現れています．

Turbo Cは，現在までに表4のようなバージョンが販売されています．

価格はVer. 1. 0が99. 95ドルでしたが，これも値上げされ，Ver. 2. 0では149. 95ドルとなっています．日本版は

表4：Turbo Cのバージョン

バージョン	発売年		おもな特徴
	US版	日本版	
V1.0	1987	—	統合環境，低価格，高速コンパイル．
V1.5	1987	1988	全体的な機能向上．
V2.0	1988	1989	全体的な機能向上．
C++V1.0	1990	1990(予定)	C++化．

当初どこが販売するかで一騒動ありましたが，結局マイクロソフトウェアアソシエイツ(MSA)社がVer. 1.5以降を販売しています（その後，新しく設立されたボーランドジャパン社が販売しています）. 日本版の価格はVer. 1.5のときは19,500円（MSA版）で，Ver. 2.0では29,800円となっています．また，Ver. 2.0以降ではTurbo DebuggerとTurbo Assemblerを含めたTurbo C Professionalというパッケージもあります．

Quick C
Microsoft社

Microsoft Cへの入門用処理系

Turbo Cの登場に危機感をもったMicrosoft社では，対抗戦略としてQuick Cを発売しました．このコンパイラは，オブジェクトコードのレベルでMicrosoft Cとの互換性をもっており，実際，ライブラリは共通のものを使用しています．コンパイル速度はTurbo Cに匹敵する高速性をもっていますが，最適化ではあまり優れてはいません．しかし，Ver. 1.0の統合環境では，ミディアムモデルしか使えないという制限があり，またリンクにはMS-LINKを使うためにどうしてもTurbo Cのリンク速度にはかなわず，総合的な評価はTurbo Cにいまひとつおよばないものでした．

もっとも，Microsoft Cのユーザーにとっては，デバッグ時には高速なQuick Cを使い，最終的なプログラムのコンパイル時にMicrosoft Cを使うという使い分けができるため，ツールとしての利用価値は大きなものがありました. 実際, US版のMicrosoft C Ver. 5.XではQuick Cを標準添付しており，こういった使い方が広く行われています．日本版では諸般の事情から，Microsoft Cとは別のパッケージになっています．

現在のVer. 2.0では，統合環境でもヒュージモデル以外のすべてのメモリモデルが使用可能になり，さらにインラインアセンブリ機能など，Microsoft C Ver. 5.1よりも進んだ機能ももっています．Microsoft C Ver. 6.0では同等品がQuick Compilerとして組み込まれましたので，もはやQuick Cは添付されていません．

Quick Cは，現在までに表5のようなバージョンが販売されています．

価格は，Ver. 1.0～Ver. 2.5まで99ドルと据えおかれています．日本版はマイクロソフト社が20,000円で販売しています．このほかに，高速なアセンブラを添付したQuick C with Quick Assemblerというパッケージもありますが，日本版は販売されていません．

Lattice C
Lattice社

根強い人気の元業界標準

Lattice Cは，MS-DOS上では最古参の部類に入るコンパイラです．MS-DOS上での最初のバージョンは1983年ごろに登場しました．MS-DOS自体の発売が1981年ですから，あまり時間をおかずに登場したといえます．これは，このコンパイラが元来マルチターゲットのクロスコンパイラであったことが力になっています．つまり，Portable Cと同様に，さまざまなCPUへの移植が容易なように設計されていたということです．

発売当初から，コンパイルが比較的高速でベンチマーク性能もまずまずであったことから人気が高く，Microsoft社がOEM採用したこともあって，事実上業界標準の処理系となっていました．しかし，1985年のMicrosoft C Ver. 3.0の登場以降は業界標準の座を追われ，さらに1987年のTurbo Cの登場でかなりユーザーを奪われ，現在に至ってはあまりパッとしない存在になっています．1986年にはMicrosoft C対抗版としての Lattice C Ver. 3.0を発売しましたが，結局は性能的に凌駕するまでいかず，1987年のVer. 3.2ではMS-WINDOWS対応を試みましたが，結局撤退するなど，なかなか厳しい経歴を背負っています．しかも，このVer. 3.2の後のバージョンが奇妙な混乱を起こしています．US版と日本版でバージョン番号が異なるのです．日本版ではこの後Ver. 4.0，Ver. 4.1を発売して今日に至っていますが，US版ではこの後Ver. 3.3を経由して1989年に突然Ver. 6.0となっています．もっとも，

表5：Quick Cのバージョン

バージョン	発売年		おもな特徴
	US版	日本版	
V1.0～V1.1	1987	1988	統合環境，低価格，高速コンパイル
V2.0	1989	1989	全体的な機能向上.
V2.5	1990	未定	全体的な機能向上．Microsoft C V6.0互換.

表6：Lattice Cのバージョン

バージョン	発売年		おもな特徴
	US版	日本版	
V1.0～V2.15	～1985	～1985	日本語は漢字プリプロセッサで対応.
V3.0	1986	1986	MS-DOS 3.X対応．オブジェクトサイズの縮小.
V3.1	1986	1986	各種機能強化.
V3.2	1987	1987	near，far等の拡張キーワードをサポート.
パーソナル	—	1987	V3.1相当の低価格版.
V3.3	1988	—	OS/2対応．オプチマイザの強化．エディタ，デバッガCodeProbeを添付.
V4.0	—	1988	オプチマイザの強化．デバッガLSDを添付.
V4.1	—	1989	OS/2対応．オプチマイザの強化．デバッガCodeProbeを添付.
V6.0	1989	未定	オプチマイザの強化．ANSI対応の徹底.

このVer. 6. 0はかなり高機能かつ高性能なコンパイラで，今後の巻き返しも期待されます．

Lattice Cは，現在までに表6のようなバージョンが販売されています．

価格は，Ver. 3. 2までは500ドルと比較的高価でした．その後徐々に値下げし，Ver. 6. 0では250ドルとなっています．日本版はVer. 3. 0までが198, 000円で，Ver. 3. 1以降は98, 000円です．パーソナルは19, 800円と，旧バージョンの1/10の値段で話題になりました．なお，DOS EXTENDERと組み合わせたLattice C/EXなど，複合化したパッケージもいろいろと販売されています．

Zortech C++
Zortech社

C++時代の先駆者

1987年に突如彗星のように現れたDatalight Cというコンパイラがありました．Datalight社から発売されたこのコンパイラは，低価格かつコンパイルが高速で，しかもグローバルオプチマイザの導入による優れた最適化能力から，ある程度の人気を得ることに成功しました．このコンパイラはWalter Brightという名のプログラマがほとんどひとりで開発したもので，1988年に彼がZortech社に移籍するとともに，C++化してZortech C++として再発売されています．

コンパイラそのものの構成は非常にLattice Cに似ています．最適化能力では最近のMicrosoft CやWATCOM Cなどに，もはや追い抜かれてしまった感がありますが，MS-DOS上では初のネイティブコードを直接生成するC++コンパイラとして高い評価を受けています．

日本版はVer. 1. 0はマイクロソフトウェアアソシエイツ(MSA)社が販売していましたが，Ver. 2. 0からは販売を停止しており，現在日本版を販売するところがない状態になっています．MSA社が取扱いを止めたのは，同社がTurbo C++を扱うことになって製品としての競合が起こることになったためで，やむを得ないのですが残念なことです．

LSI C-86
エル・エス・アイ ジャパン社

現存する唯一の国産処理系

パソコン上では，一般のアプリケーションプログラムでもそうですが，Cコンパイラについても残念ながら欧米で作られたものが主流で，国産の処理系というものはほとんどありません．そういう中で，現存する唯一の国産処理系といえるのがこのLSI C-86です．

標準でROM化ツールやライブラリのソースを添付し，最適化などにも優れた性能を示すのですが，派手な特徴がなく，全体的に地味な印象のためか人気はいまひとつというところです．

1987年にVer. 3. 0が発売され，1990年現在，Ver. 3. 2が最新版です．価格は48, 000円です．1989年からスモールモデルに限定した評価用のセット（試食版）を主要な国内BBSや雑誌の付録のフロッピーなどで配布しており，この点は非常に高く評価できます．

少数派のC処理系
―米国では新しい動きも

現在，日本で広く使われているのは，以上の5つの処理系です．この他にも次のような処理系が使われていますが，現在では小数派になってしまいました．

Aztec C86
(Manx Software Systems社)
DeSmet DC88
(C-Ware社)
HI-TECH C
(HI-TECH Software社)
Mark Willams C, Let's C
(Mark Williams社)
Optimizing C86, C86PLUS
(Computer Innovations社)
Power C
(Mix Software社)
Whitesmith C86
(Whitesmiths社)

日本では最近は，Microsoft CとTurbo Cが2大勢力となってしまったため，なかなか新しい処理系で対抗しようとする動きが少なくなってしまいました．しかし，米国ではまだまだ新しい動きがあります．代表的なのが次のふたつの処理系です．

WATCOM C
WATCOM Products社

カナダから現れた新星

WATCOM社は昔からFORTRANコンパイラなどを開発してきた会社ですが，1988年頃から急にCコンパイラで注目されるようになりました．当時のWATCOM Cのバージョン番号はVer. 6. 5と大きな数字ですから，かなり以前から販売されていた処理系だと思われます．

このWATCOM CはMicrosoft C Ver. 5. 1よりも最適化において優れた性能を示すという点からかなり高い評価を得ています．無論，エディタやソースレベルデバッガなどのツールも充実しています．1990年現在ではVer. 8. 0が最新版で，OS/2やWINDOWS, DOS EXTENDERにも対応するなど，非常に高い開発力をもっているようです．

しかし，残念なことに今のところ日

本版を販売する動きは見られません．コンパイラそのものもIBM PCへの依存度が高いようです．

Topspeed C
JPI社

因縁の本家Turbo C？

JPI社は，Borland社をスピンアウトした連中が設立した会社です．このTopSpeed Cについては，因縁めいた話が伝わっています．前述の通り，Turbo Cは最初からBorland社によって開発されたものではないという話ですが，Borland社では当初は完全に自社開発の予定で，実際Cコンパイラを開発していたのだそうです．しかし，どういう事情かはわかりませんが，結局途中から完全自社開発を諦め，前述のように他社から買取ったコンパイラを元にTurbo Cを作り上げたようです．そして，開発を中断したオリジナルTurbo Cともいえるコンパイラは，結局スピンアウトした連中からTop-Speed Cとして発売されることになったのです．無論，このエピソードも両社から正式にコメントされたものではなく，やはり単なる噂話なのですが…．

まあ，ともかくTopSpeed Cはなかなか高性能なコンパイラです．ベンチマーク性能ではMicrosoft C Ver. 5.1にはおよばないようですが，Turbo Cよりはよい性能を示します．逆にコンパイル速度は比較的高速ですがTurbo Cにはおよばないようです．ライブラリ関数はMicrosoft CとTurbo Cの両方に互換を取るという大胆なことをやっています．このため，同じ機能の関数がかなりだぶって提供されています．グラフィックライブラリもTurbo C（BGI）互換のものとMicrosoft C互換のものがあります．

MS-DOS上でダイナミックリンクライブラリ（DLL）の機能をサポートしたり，OS/2やWindowsへも対応するなど，かなりの開発力を見せています．統合環境やソースレベルデバッガなども充実しています．しかし，この処理系も残念なことに，今のところ日本版を販売する動きは見られません．コンパイラそのものもIBM PCへの依存度がかなり高いものです．

また，DOS EXTENDERやOS/2 2.0による80386用のプログラムを開発するために80386のコードを生成できるコンパイラもいくつか現れてきています．代表的なのがMetaware社のHigh C386ですが，WATCOM C386やNDP C-386といった処理系も登場してきています．

将来への展望
GUIへの対応とC++化

今後も，最低数年間はMS-DOSがパソコンの標準的なOSであり続けると予想されます．無論，前述のようにMS-DOSではメモリ空間の制限が厳しい状況ですし，グラフィカルユーザーインターフェース（GUI）の標準化を求める時代の流れからは，いずれWINDOWSやOS/2が標準的なOSとなっていくものと思われます．

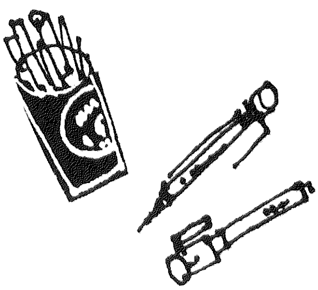
現時点では，OS/2に対応している処理系は，Microsoft C，Lattice C，TopSpeed C，WATCOM Cの4つで，またWINDOWSに対応しているのはMicrosoft C，TopSpeed C，WATCOM C，Zortech C++の4つですが，他社も次々に対応してくるでしょう．

また，あわせてC++化も進むものと思われます．C++はポストC言語の本命と目されているからです．現在のところ，ネイティブコードを生成するC++処理系は，Turbo C++とZortech C++のふたつだけですが，他社もいずれ対応してくるものと思われます．

MS-DOS Ver. 5.0について，この原稿を書いている時点では詳細は不明ですが，DOS EXTENDERについて何らかの標準化がなされるものと予想されています．そうなると，80386コードの生成なども課題となってきます．

もちろん，コンパイラの細かい改良や，ツールの充実などさまざまな改良は今後も続き，当分の間はCまたはC++がパソコンの標準開発言語の地位を維持するものと思われます．

今回は，MS-DOS上のCコンパイラをとりまく現在の状況について説明することに終始してしまい，あまり「実践的」な内容ではなくなってしまいました．ベテランの方や，具体的なプログラミング技術を期待していた方には肩透かしになってしまったかもしれません．次回からはMicrosoft CとTurbo Cの特徴を細かく検討することによって，MS-DOS上での具体的なプログラミングについても述べていきますのでご期待ください．



98活用アセンブラ技法

ビデオシステム編 第1回

MASM・TASMによる構造化マクロライブラリの製作

星野 操

この連載では，ハードウェアの機能を100%引き出すためのアセンブラプログラミングについて解説を行っていく．具体例としては，PC-9801シリーズのビデオシステムにバックグラウンドカラーと二重罫線を付加するプログラムを予定している．初回は，アセンブラプログラミングの準備として構造化マクロの紹介を行った．

教養としてのアセンブラ

一人前のプログラマになるために一度は通らなければならない必修科目

編集部の方から依頼されて，本誌でアセンブラでのプログラムについての連載をもつことになりました．

現在では，プログラムをアセンブラで書かなければいけない状況というのは，けっして多くはないと思います．とくに，プログラムを頭から終りまでアセンブラで書くということはめったにないでしょう．筆者もいくつかのプログラムをオールアセンブラで書いたことはありますが（そのうち，ひとつはかなり大きなプログラムとなりました），現在ではそれほどアセンブラべったりのプログラムを書くことはあまりありません．

そもそも，高級言語に対するアセンブラの利点は，（速度・サイズの面で）効率の高い記述ができる，高級言語が本来カバーしていないCPUに密着した記述（CPU自身のコントロールなど）ができるということになるでしょうか．

しかし，現実では速度の面で極限までの効率を求められることはあまりないでしょうし，MS-Cによって導入されたfar/nearキーワードによってメモリ空間のアクセスは非常に容易になり，Turbo Cの画期的な（?）interruptキーワードによって，最後までアセンブラ記述の要求されていた割込みハンドラまでもCのみで記述可能になってしまいました．

加えてTurbo Cの邪悪な（:-）レジスタ疑似変数を併用すれば，およそCで「記述できない」プログラムはほとんどなくなってしまうでしょう．さらに，インラインアセンブラやTurbo Cの邪悪な（:-）__emit__疑似関数を使用すれば，記述不可能な領域はかぎりなく0に近づいてしまいます．

ただあまりに変な技巧にこると，かえってプログラムの可読性を悪くすることもあります．Turbo Cなどで導入された技巧は，あくまで「アセンブラでしか書けなかった」記述をCで可能にしたものです．ですから，アセンブラでの書き方がわからなければ，やはりCでも書けるようにはなりません．

また，高級言語では使用するシステムライブラリの関係から，どうしてもプログラムサイズが大きくなる場合が多いのです．通常のプログラムではあまりメモリサイズは気になりませんし，やがて80386のプロテクトモードでも使用するようになれば，メモリサイズはほとんどまったく制約をもたらさなくなるでしょう．

しかし，現状では常駐プログラムやデバイスドライバなどの用途において，作成されたプログラムのサイズが大きな意味をもってくることもあります．さらに画面処理などの，速度に関して，とくにクリティカルである類のプログラムでは，やはりアセンブラに

よるオプティマイズが好ましくなります．

何から何までアセンブラで書こうというのは，先に書いたように現実的ではなくなってきています．可能な範囲では，やはり高級言語で記述するべきでしょう．ただ，「教養として」でもアセンブラを知っていれば，実際にアセンブラが有効に使用できる場面において，その判断が適切にくだせるようになると思います．

さらに，CPUの挙動自体を理解するためには，やはり機械語，そしてアセンブラを習得しておく必要があるでしょう．これは，とくにハードウェアに密着したプログラムを書く場合や，システムカーネルなどを書く場合には必須であるといえます．

以上のように，やはりアセンブラは一人前のプログラマにとっては，避けて通ることのできない必修科目であり続けるのでしょう．

MASM5.1, TASM1.0

高級言語とのリンクに深い配慮がなされたアセンブラ

とはいっても，筆者にはアセンブラの入門記事を連載することはできません．ですから，ある程度はアセンブラでプログラムが書ける程度のレベルの人を対象にします．ここではなるべく多くのアセンブラソースを掲載しながら（ページ稼ぎをしつつ　:-），それに対する解説を加えることで，筆者のもっているつたないプログラミングテクニックを掲載する，ということになるでしょう．

本連載で対象とする処理系は，TASM1.0およびMASM5.1とします．掲載するプログラムは，両者でアセンブル可能であることを（できるだけ）確認し，なんらかの問題がある場合にはコメントするように心がけるつもりです．

MASM5.1からは，高級言語に対する深い配慮がなされるようになり，とくに高級言語のサブルーチンをアセンブラで書く場合には，ほとんど何も考えなくてよいほどになっています．これは，以前のバージョンに比べて大きな利点であるといえるでしょう．ですから，積極的に新しいバージョンを使用するべきであると思います．古いバージョン，たとえば古いDOSに添付されていたバージョンなどを使用されている方は，ぜひ新しいものを入手されるようにお勧めします．

筆者は好んでこれらの新しい機能を使用しています．こうすると，レジスタ使用の最適化，たとえば関数に対するレジスタ渡しや関数呼出しをはさんでのレジスタの内容の保存性など（アセンブラ特有な記述）に目をつぶって，スタック内に確保したローカル（メモリ）変数を多用することになりがちです．このため，サイズ・スピードの面でのアセンブラのメリットが薄れてきてしまいます．

ここらへんのトレードオフはむずかしい問題ですが，やはりプログラムでいちばん大切なのはまず「動くこと」．そして動いてからチューンナップすればよいと考えれば，最初は多少冗長でもわかりやすく書いておいたほうがよいと思います．

なお，OPTASMについては基本的に対応しません．筆者が処理系をもっていないことに加えて，生成コードの質が異なってくるためです．たとえば，本号に掲載する構造化マクロなどは，OPTASMでは正常に処理できないでしょう．これは，本マクロが前方参照のjmp命令が3バイトのコードを生成することを仮定していたりするからです．OPTASMでは，マルチパスの最適化機能によってこの仮定が成立しません．ただし，これらは主として条件付きのlongジャンプを実現するために使用されているので，OPTASMを対象として書き直せば，かえって記述は簡潔になると思います．

作成するプログラム

PC-9801でバックグラウンドカラーと二重罫線を実現する

アセンブラでのプログラミングといっても，何も課題がないとやりにくいので，本連載ではPC-9801の画面処理を対象とすることを考えています．

PC-9801ではバックグラウンドカラーの表示ができません．また，罫線キャラクタについてもシングル罫線しか用意されていないので，メニュー画面などを表示させた場合にどうしても画面が貧弱になってしまいます．

そこで外字を使用して二重罫線を，グラフィックを使用してバックグラウンドカラーの表示を可能とすることを試みます．これらの機能をCの関数の形で呼び出せるようにするか，あるいは常駐プログラム形式にしてエスケープシーケンスでコントロール可能となるようにします．もちろん，画面の部分取得や部分スクロールなども関数形式でサポートすることになるでしょう．

なお，上記のインプリメントには，ハードウェアの都合上PC-9801VM以降の機種を想定することにします．具体的には，256文字の外字を使用でき，GCを搭載しているシステムということです．

以上のようなプログラム例を紹介することで，高級言語の関数を記述する方法などのアセンブラの一般的な技術

を紹介することに加えて，PC-9801のハードウェアのアクセスのしかたなどについてもある程度紹介することにします．

MASMのマクロ機能

演算機能・制御構造・マクロ定義マクロで強力なマクロ機能を実現

MASMは，かなり強力なマクロ機能をもっています．これは，マクロの中で演算機能が使用できること，制御構造が使用できること，マクロ定義マクロが定義できることによっているといえます．

簡単な文字列の置き換えなら，CPPのようなプリプロセッサで実現することができます．実際，yacc/lexで簡単なアセンブラを作り，マクロ機能についてはCPPで補うというような例も雑誌で紹介されたことがあります．

しかし，このCPPで実現されるマクロは，上記の3条件がすべて満たされていないことから非常に貧弱なものとなってしまいます．多様な記述をするためには，演算評価を伴った制御構造は必須ですし，下手に使うとわけがわからなくなりますが，マクロ定義マクロはかなり有効なものです．

具体的には，後方参照の条件ジャンプが考えられます．8086では条件ジャンプはショートジャンプですから，ジャンプ先との距離を考えて，必要なら反対条件のジャンプと無条件nearジャンプとを組み合わせる必要があります．

自動的にどちらかを選択するマクロ命令を定義しようとした場合に，ラベルと現在のロケーションとのアドレスオフセットを演算し，その値の大小によって生成する命令コードを変える必要があります．これは，MASMなら苦もなく実現できることですが，CPPなどの置き換えマクロではどうやっても実現できません．また，デフォルトのロケーションカウンタに対する概念も非常に便利です．

これらの機能を使用すれば，かなりいろいろなことがマクロで記述できます．

一般にマクロというと，複数のアセンブラ命令を1命令で記述するような用途がおもになっています．たとえば，特定のI/Oポートにアクセスするために，DXに対する値の設定と入出力命令をひとつの命令にまとめてしまうような場合などです．

このような用法は，「使い捨て」マクロに多く，使い捨てマクロでは，用法などが限定されているので，さほどきちんとしたドキュメントを作成する必要はありません．しかし，マクロの記述能力はこのようなシンプルな用法をはるかに超えているので，より汎用性のあるライブラリを構築することも可能です．そのような場合には，必ずきちんとしたドキュメントを残しておくようにしないと，あとで何がなんだかわからなくなることはまちがいありません．

構造化マクロ

ローカルラベルの管理を簡略化するためにマクロを導入

本連載では，最初にアセンブラに制御構造を導入するマクロを紹介します．本連載で掲載するプログラムは，以後全面的にこれらのマクロを使用することになります．

アセンブラの記述で非常に面倒なことのひとつは，ローカルラベルの定義です．裸のアセンブラでは制御構造の記述がサポートされていませんから，どうしても分岐命令を多用する必要が生じます．分岐先は通常ラベルで指示するので，ローカルラベルが増えてしまい，その管理がたいへんになるのです．

MASMなどでも新しいバージョンではローカルラベルの機能が導入されましたが，ネストされたループを記述する場合などには力の弱さが目だってしまいます．このため，この欠点を補填するマクロというものがいくつも発表されています．古いところではアスキーのA-MACROSというのがありますし，そのあとにも多くのマクロがネットワーク上や雑誌上で発表されてきました．

こういうものは，だれもが望んでいるわけなのですが，ひとりひとりの求める方向には少しずつ違いがあるため，多くのバリエーションが作られてきました．ここで，さらに新しいバリエーションを公開し，それを強制的に使うということに対しては，ひけめを感じないでもないのですが，筆者が比較的長く親しんできた構造化マクロだということで大目に見てやってください．そして，このマクロ自体を使うことはなくとも，マクロの書き方などを参考にしていただければ幸いです．

このライブラリのオリジナルは，2年半ほど前に書きはじめたものです．自作のアセンブラプログラムのラベル管理がそろそろ限界に達してきたため，記述を簡略にするためにこのライブラリを使用し，全面的に書き換えることにしたのです．

しかし，当時使用していたMASM(5.1だと思う)では，不可思議なphase errorが大量に発生し，ほとんど実用にたえませんでした．空行を入れたりすることで回避できるのですが，プログラム自体を変更するとエラーが発生す

ワンラインNEWS
韓国の政府系研究教育機関，韓国科学技術院は並列処理による高速コンピューターを開発した．32個のプロセッサを連結し64 MFLOPSの性能を実現した．

る位置が動いてしまい，はじめからやり直しになるのです．たぶん，アセンブラのマクロハンドリングに問題があったのだと思いますが，結局同等の仕様のプリプロセッサを作成することで回避しました．

ただし，先ほど述べたようにプリプロセッサでは不十分な点もどうしても出てきてしまいます．ですから，必要最小限のマクロを定義し，プリプロセッサはそのマクロを使用した出力をする，という手段をとっていました．

プリプロセッサを使用するもうひとつの問題点は，実際にアセンブルされるソースが，記述したそれと異なっているということです．たとえばアセンブラでエラーが発生した場合には，一度プリプロセッサの出力したソースを参照しエラー場所を確認してから，ソースを修正する必要が生じます．一般に，これらのファイル間で行番号が異なってくるので，かなりめんどうなわけです．

さらに，最近はやりのウィンドウタイプのデバッガを使用する場合でも，トレースなどの単位行がソース記述のそれか，プリプロセッサ出力後のそれかで使い勝手に大きな差が出てきてしまいます．

TASMでは比較的これらが安定していることが期待されたので，今回仕様をなぞってライブラリを全面的に書き直したところ，比較的よい結果が出ました．非常に大きなプログラムの場合はどうなるかわかりませんが，さしあたって新しいライブラリはMASM 5.1でもなんとか動いているようです．もちろん，不安がまったくないわけではないのですが……．

```
;-------------------------
;条件構造化マクロ
; condx: condition structure macros
;
; $Log: RCS/condx.inc $
# revision 1.1 TAN 90/09/18 09:13:21
# Initial revision
#
;-------------------------

        .xlist
        .xall

;-------------------------
;デバッグオプション
; debugging option.
; $xlist/$list are used in this library to suppress macro
; expansion.
; set 1 to $debug to expand all macro contents, for debugging
; this lib itself.

$debug  =       0               ; 1 for lib debugging
        if      $debug eq 0
                $xlist  equ     <.xlist>        ;; normal
                $list   equ     <.list>
        else
                $xlist  equ     <>              ;; don't suppress macro
                $list   equ     <>
        endif

;-------------------------
;エラーメッセージ出力の調整
; for console error message output.
; %out command outputs messages every pass on masm.
; so, if2-endif used to suppress outputs on pass1.
; while, tasm outputs messages first, only one time.
; enclusing %out with if2-endif, makes error.
; so, following macros are used.

        ifdef   ??version
                ??TASM  equ     1
                $if2    equ     <>              ;; for tasm
                $endif2 equ     <>
        else
                ??MASM  equ     1
                $if2    equ     <if2>           ;; for masm
                $endif2 equ     <endif>
        endif

;=========================
;
; STRUCRTURE MACROS
;
;-------------------------
;構造コントロール用変数の準備
; stack works
; $stack:stack level counter
;        show the nest level
; $num  :for sequential number
;        incremented by every structure macro
; $val  :temporaly numerical macro
;        used for evaluation.
; $temp :temporaly literal macro.

$stack  =       0
$num    =       0
$val    =       0
$temp   equ     <>

;-------------------------
;ネストチェックid
; structure nest checking id
; used in $push/$pop macro
; at the end of a structure, id is poped out.
; if it is not one pused at the top of paired
; structure, error occurs.

IF_PTR  equ     1
BEGIN_PTR       equ     2
WHILE_PTR       equ     3
DO_PTR  equ     4
MC_PTR  equ     5

;-------------------------
;逆条件のジャンプ命令を生成
; create reverse condition jump macro.
; example, $njnz is equall to jz.
; and $njz is equall to jnz

        irp     x,<z,c,s,a,ae,b,be,g,ge,l,le>
$nj&x   equ     < jn&x>                 ;; additional space is for tasm
$njn&x  equ     < j&x>                  ;; also for tasm
        endm
```

マクロライブラリの基本構文

基本的にforth的な構文をサポート

さて，本マクロライブラリは，基本的にforth的な構文をサポートしています．forth系列の構造は，ブロックがはっきりしているため，比較的作りやすいという利点があります．制御構造は分岐と繰り返しからなり，繰り返し条件の判断は，終端と開始時に行う2つの構造が用意されています．また，回数を指定した繰り返しも可能です．制御構造マクロは，@からはじまる名前で定義されています．

このマクロの特徴として，条件分岐・繰り返しマクロの引数には分岐条件のみが指定され，条件を生成するコマンド列はとられないことがあります．多くの構造化マクロでは，ひとつ程度の命令を引数にとり，

```
$if      <cmp  ah,al> ,z
```

というような記述を可能としているのですが，このマクロでは単に

```
@if      z
```

と記述されるだけです．条件を生成する命令は，@ifに先立って記述（実行）されていなければいけません．いくつかの命令を実行させるようにすることは簡単なことです．ただ，forth的な考えではそれは美しいとはいえなくなりますし，アセンブラでの記述性が不当に落ちる可能性があります．「分岐条件の使い回し」のようなことができなくなるからです．まあ，このあたりの問題はポリシーの問題ですから，どうなるものでもないでしょう．

もうひとつ，分岐系の命令では前方参照による分岐を行います．8086では先に述べたように条件分岐命令では分岐範囲に制限があります．ですから，ひき続く記述の多少によって生成コードを変える必要があります．この自動判別は（OPTASMでも使わないかぎり）困難ですので，マクロ引数で指定してやることにします．

@ifなどのマクロの第2引数にfarなどの文字列をとってやると，生成されるコードは長い条件ジャンプを想定したものとなります．通常はブランクにしておき，プログラムを書きたしてエラーが出るようになった次点でfar記述を付加してやればよいでしょう．

さて，このような記法で生じる困難のひとつは，「elseif」です．

pascalやCなどの言語の場合，条件分岐文は，次のひとつの文のみを実行するようになっています．ですから，else ifと簡略に多重分岐が記述できるのです．これに対して，ifの終端を明示するタイプの言語では，多重分岐をするとどうしてもネストが深くなり，最後はendifの山になってしまうのです．実際，MASM自身の制御構造であるif-else-endifなどについても，以前はelseとifの多重ネストになり，最後はendifの山になってしまいました．

これに対する対策のひとつは，elseとifを一体化しelseif系列の命令を作ることです．実際，MASMも5.1(?)からこのelseif系の疑似命令が導入されたため，あまり深いネストを作らなくてすむようになりました．

しかし，本ライブラリのような記法をとっていると，elseifの判断条件をする命令を記述する部分が（elseif自身の引数にとれないので）なくなってしまいます．このため，簡単にelseifを導入することができません．

ここでは，@elseのあとに@e_ifという特別なifを導入することで，なんとか解決をはかっています．つまり，@elseと@e_ifの間の命令で条件を作成し，@e_ifの引数によって分岐するわけです．もちろん，@elseと@e_ifの間には複数の命令が記述できるので，直接分岐に関係ない命令を書くこともできます．これによって，記述性をアップすることができます．

以上をまとめると，条件分岐系のブロックは次のような構造になります．

```
@if      条件，[範囲]
         命令
@else
命令
@e_if    条件，[範囲]
         命令
@else
         命令
@endif
```

@ifブロックは，必ず対応する@endifでターミネートされます．@ifには，先に述べたように条件を引数とし，ジャンプ先のレンジを示す引数がオプションとして与えられます．

このブロックの中では@elseを使うことができます．@elseは@ifまたは@e_ifと，@e_ifまたは@endifの間にひとつだけ記述することができます．@e_ifは，@elseと，@elseまたは@endifの間にひとつだけ記述することができます．もちろん，これらの構造はネスト可能です．

@e_ifと書くべきところを@ifと書いてしまうことがよくありますが，そのような場合にはネストエラーが発生します．

```
;
;-----------------------------
;文字変数・数値変数に値をセット
; value setting macro
; usually, used to evaluate arguments
; $set is for setting literal value.
; $setn is for setting numerical value.
; also, $inc/$dec are defined.
;
$set    macro   var,arg
var     equ     <arg>
        endm

$setn   macro   var,arg
var     =       arg
        endm

$inc    macro   arg
arg     =       arg+1
        endm

$dec    macro   arg
arg     =       arg-1
        endm

;-----------------------------
;反対条件ではないジャンプ
; create jump on non reverse condition
; argument label is 'evaluated'.
; following description
;       $jcond  z,$1%$val
; will create
;       jz      $112
; if $val contents 12.
; use argument 'mp' as cond(ition), to create non
; conditional jump to a label that must be evaluated.
;
$jcond  macro   cond,label
        $list
        j&cond  label
        $xlist
        endm

;-----------------------------
;反対条件のジャンプ
; create jump on reverse condition
; use $njxx literal macro for reverse condition.
; first, create jmp command literal macro,
; then expand it.
;
$jncond macro   cond,label
        $set    $temp,$nj&cond          ;; create literal macro,
        $list
        $temp   label                   ;; and evaluate it
        $xlist
        endm

;-----------------------------
;ラベル定義
; label defining macro
; also used to evaluate numerical macro.
;       $label  $1%$val
; will create
; $112:
; if $val contents 12.
;
$label  macro   label
        $list
label:
        $xlist
        .xcref  label                   ;; suppressing cross reference list output
        endm

;-----------------------------
;ブロックidのプッシュ
; pushing id macro
; argument: cheking id
; $iXX = structure id
; $nXX = sequential number
; $num, $stack both incremented
;
$push   macro   id
        $setn   $i%$stack,id            ;; push id value
        $setn   $n%$stack,%$num         ;; push sequence number
        $inc    $num                    ;; and update variables.
        $inc    $stack
        endm

;-----------------------------
;ブロックidのポップ
; poping id macro
; argument: cheking id
```

多重条件のサポート

複数のフラグの状態のand/orの結合として表現

このライブラリでは多重条件を（かなり無理に :-) サポートしています．現在では分岐系のみのサポートですが，繰り返し系をサポートすることはさほど困難ではありません．

多重条件は複数のフラグの状態のand/orの結合として表現されます．現在では，やはり簡略記述がされているので多重の多重条件が記述できず，1段の積和しか記述できませんが，@condの記述を変えることで対応することができるはずです（以前のバージョンではこれが可能となっていました）．

具体的には，@mcと@ifの間が多重条件記述ブロックとなります．この中には，複数の@condマクロが記述され，それらすべてが満たされたときに@ifのあとのブロックが実行されることになります．なお，@mcに対応する@ifの条件引数は空白とします．@condマクロは前方参照をするので，レンジ指定引数を与えることができます．

@condがひとつでも満たされないと，そこで多重条件ブロックの実行は中断され，制御は対応する@ifブロックの@elseまたは@endifのあとに移ります．ただし，多重条件ブロックの中に@orが表れている場合には，@condが満たされないと次に表れる@orのすぐあとに制御が移ります．一方，そこまでの@condがすべて満たされていた場合にすぐに@ifのあとに制御が移ります．

これら@condと@orは，イメージ的にはCの条件制御演算子，&&と||にきわめて近いものと捉えていただければよいでしょう．プログラム例を紹介するところで，具体的に説明することにします．

なお，これらの多重条件記述はジャンプが重なる（分岐先が分岐命令である）ことがあるために，あまり効率的ではありません．もしもスピードなどにクリティカルな部分であれば，記述を改良する必要があります．

繰り返し構文

3つの@beginループと@do-@loopを装備

通常の繰り返し系列では，ブロックは@beginではじまります．@againでブロックが閉じられた場合，無限ループが形成されます．ここから抜け出すには，@breakまたはretなどを使用します．

@untilマクロは条件引数をひとつとり，その条件が満たされるまでループを繰り返します．Cの制御構造でいえばdo {} while () に近いものです．

もっとも柔軟な繰り返しブロックは，@begin-@while-@repeatで形成されるものです．まず@beginと@whileの間の命令が実行されます．そして@whileの引数である条件がチェックされ，それが満たされれば@repeatまでの間が実行されます．@repeatに達すると，制御は再び@beginに移ります．一方，@whileの条件が満たされなければ，制御はすぐに@repeatのあとに移ります．

@beginと@whileの間に，条件生成のための命令だけを置けば，これはCでいうwhile () {} 構文に近いものとなります．ただし，実際にはそれに限らず自由な記述が可能なので，より複雑な制御もできます．具体的には，for文の再初期化にあたるようなものを置くことも可能なのです（もちろん，最初に一度実行されてしまいますが）．なお，@whileマクロは前方参照を行うため，レンジ引数を取ることができます．

このほかに，一定回数のループを形成するための@do-@loopブロックがあります．このバージョンでは簡略化したために，ループステップの指定やCXを使用した場合のオプティマイズなどは省略されています．残念ながら，全体に効率はあまりよくないものです．

@doは最低3つの引数を取ります．最初はループカウンタであり，レジスタか変数を取ります．次がループの終値，3番目がループの初期値です．4番目はレンジ指定の引数です．

forthの仕様に習い，ループの終値はそれ自体ループカウンタの値として実行されることはありません．たとえば，10回のループを組みたい場合には，

```
@do     cx,10,0
        命令
@loop
```

という具合にします．cx＝0～9の間でループは実行されます．初期値，終値はレジスタや変数でもかまいませんが，ループカウンタとの双方が変数になることは避けなければいけません．

ループからの脱出などのために，@breakと@continueが用意されています．@breakはそのすぐ上のループブロックの終端に制御を移します．@continueは，@doまたは@beginのすぐあとに制御を移します．Cのそれと同様に考えていただけばよいでしょう．

ワンラインNEWS
日本語ワープロのかな配列キーボードの改良を進める日本入力コンソーシアムはキーの基本配列の名前を「NICOLA」に決定した．

以上が，本ライブラリの主要なマクロの説明です．ある程度のコメントがマクロ中にはいっているので，詳しくはそちらのほうを参照してください．また，おいおい使用例も紹介することになると思います．

マクロライブラリ

ライブラリ内で使用されるマクロ・ラベル

$debug/$xlist/$list

さて，ではこのライブラリ「自体」の説明を少しすることにしましょう．このライブラリでは，プログラム中で使用するマクロなどは@ではじまり，ライブラリ内部で使用するマクロやラベルなどは$ではじまるようになっています．

アセンブラのリスティングにこれらマクロの展開結果をすべて入れてしまうと，たいへん見にくくなります．このために，必要に応じて．xlistと．listを使い，リスト出力をコントロールしています．ただし，ライブラリ自体のデバッグのためにリストが必要な場合もありますので，そのような場合は$debugの値を非ゼロにすると，このようなコントロールが行われなくなります．

実際には，マクロ内部では．xlist/．listではなく$xlist/$listという命令が使用されています．これらは，文字マクロとして定義されています．このようにマクロを使用すれば，ひとつのコマンドを別名で使用することもできます．

```
; if top of $iXX stack is different to argument id,
; error occurs.
; $val numerical macro has the value of tos of $nXX stack.
; this value will be used to create label.

$pop    macro   id
        if      $stack eq 0             ;; first check stack underflow
                .err                    ;; cause user error
                $if2                    ;; this will act on masm only
                        %out    COND Error: stack underflow
                $endif2                 ;; this line also.
                exitm
        endif
        $dec    $stack                  ;; stack decrement
        $setn   $val,$i%$stack          ;; get id value using $setn macro
        if      id ne $val              ;; compare id
                .err                    ;; cause error if ids are different
                $if2
                        %out    COND Error: nesting error
                $endif2
                exitm
        endif
        $setn   $val,$n%$stack          ;; then, get sequence number
        endm

;-----------------------------
;ブロックidのチェック
; checking id macro
; this macro never push/pop stack. but sample their contents.
; similar to $pop macro, top of $iXX stack must be equall to
; argument id. otherwise, error occurs.
; and top of $nXX stack is set into $val variable.

$check  macro   id
        $setn   $val,$i%($stack-1)      ;; get top of id stack
        if      id ne $val              ;; and compare it
                .err                    ;; if not identical, error
                $if2
                        %out    COND Error: nesting error
                $endif2
                exitm
        endif
        $setn   $val,$n%($stack-1)      ;; return top of number stack
        endm

;-----------------------------
;ブロックidのスワップ
; swapping id macro
; argument: cheking id, new id
; if top of $iXX stack is different to checking id, error occurs.
; set new id to top of $iXX stack
; set top of $nXX stack to $val variable

$swap   macro   id,newid
        if      $stack eq 0             ;; first, stack underflow checking
                .err
                $if2
                        %out    COND Error: stack underflow
                $endif2
                exitm
        endif
        $setn   $val,$i%($stack-1)      ;; then, top of id stack checking
        if      id ne $val
                .err
                $if2
                        %out    COND Error: nesting error
                $endif2
                exitm
        endif
        $setn   $i%($stack-1),newid     ;; change top if id stack
        $setn   $val,$n%($stack-1)      ;; and return value
        endm

;-----------------------------
;多重条件定義
; multiple condition definition
; used with condition evaluating structure, such as
; @if or @until.
; @mc shows the top of multiple condition definition.
; @if or such macro shows the tail of it.
; condition satisfied, if all @cond conditions satisfied
; between them.
; if @or is used in the structure, execution condition
; satisfied if all @cond conditions between @mc or previous
; @or satisfied.
; with following example,
;       @mc
;               cmp     ax,1
;               @cond   z
;               cmp     bx,0
;               @cond   z
;       @or
```

$if2／$endif2

次に，エラーメッセージの部分のための細工があります．MASMは2パスアセンブラであるので%out疑似命令を使用すると2回メッセージが出力されてしまいます．そのため，if2を使用して2パス目だけにメッセージを出力することになるのですが，これがTASMではエラーになってしまいます．そのため，$if2／$endif2を利用し，TASMでは$xlist／$listと同様にこれに空白を定義します．

$stack/$num/$val/$temp

その次に，構造コントロール用の変数が用意されています．ここでは，主として$stackと$numが使用されます．$stackはネストレベルを格納し，ネストが1段深くなると1増えます．$numはラベル生成用のカウンタで，必要なラベル用の番号を確保するごとにひとつ増えます．このほかに，汎用(評価用)の数値マクロに$valを，文字マクロに$tempを使用します．

$i??

ネストの構造チェックには，$i??という形式の変数（数値マクロ）を使用します．??の位置には構造のネストレベルを数値で表現したものを使用し，構造を示す数値をセットします．たとえば，もっとも外側（レベル0）にifブロックがあるときには，対応する$i0にifブロックを示す1をセットすることにします．これらの値は，スタックのように使用されますから，現在のネストレベル以上の値は無効です．

$nj&x/$njn&x

次に，逆条件のジャンプ命令を生成するためのマクロがあります．ここではirpでループを回し，対応する条件(たとえばnz）の前に$njを付けたもの(たとえば，$njnz）に反転条件のジャンプ命令(ここでは，jz)を定義します．分岐の部分でひんぱんに反転条件の分岐命令を使用する必要があるのですが，それはこの文字マクロによって生成することにします．

$set/$setn

その次に，数値変数・文字変数に値をセットするマクロ（$set/$setn）などがあります．これらのマクロは，主として引数を「評価」するために使用します．MASM/TASMには多段階の文字マクロ評価を行うための演算子%が存在しますが，これでは1レベル評価した文字マクロを定義したい，というような場合に不便です．

具体的な使用例は，分岐命令を生成する$jncondに見られます．引数とする条件condの内容がnzの場合に，$njnz命令を使用したいのですが，たとえば直接

```
$nj&cond        label
```

などとしてもうまくいきません．命令の部分では，文字列の結合演算はそれ自体の意味をもち，結合結果の名前をもつ文字マクロが展開されるわけではないからです．ですから，最初に文字列結合した結果を適当な文字マクロに代入し，その値を使用する必要があります．$tempにこの値を代入するために%演算子を使用すれば，たとえば

```
% $temp   equ      <$nj&cond>
```

となります．ここで$tempが文字マクロとして定義されていなければ，問題なく$tempに$njnzという文字列が定義されるのですが，もしも$tempにすでに値が（文字マクロとして）設定されている場合には，その内容とする変数に値が設定されてしまうのです．

このような場合に，$setのようなマクロを使用すれば必要に応じて「1段階の評価」が可能となります．代入する変数名はそのまま，値である「$nj&

ワンラインNEWS
IBM社は中国・天津市にパソコンを製造するための合弁会社，天津先進信息産品有限公司を設立した．「PS/2」を日産100台を製造する予定．

cond」だけを評価させると求める結果がえられます．

同様に，$stackが0のときに$i0の値を取得する場合にも，引数評価の演算子%と$setnマクロを併用すればよいことになります．

このようにマクロを使用すると引数の1段階の評価が可能なので，ここではジャンプ命令やラベルの生成についてもマクロを使用しています．

$inc/$dec

ほかには，記述を簡単にするために数値マクロ変数をインクリメント・デクリメントするためのマクロも用意してあります．これらは，$stackや$numをコントロールするのに有用です．

$push/$pop/$check

$push/$pop/$checkなどは構造をコントロールするマクロです．ブロックのネストが1段深くなるときには$pushが呼出され，浅くなるときには$popが呼出されます．具体的には，ブロックの先頭で$pushが呼出され，最後に$popが呼出されます．また，ブロックの内部でブロックIDをチェックする場合など（@elseなど）では$checkが使われます．

$pushは引数にブロックIDを取ります．これを$i??に設定し，$n??に$numの値を設定した後$numをインクリメント，スタックレベルを1増やします．$popでは，カレントブロックの$i??の値と引数とを比較した後，スタックレベルを1減らします．また，このときの対応する$n??の値を$valに代入します．

たとえば，@stackが2で@numの値が1である場合には，@ifでは$l1への条件ジャンプ命令を生成します．こ

```
;               cmp     ax,0
;               @cond   z
;               cmp     bx,1
;               @cond   z
;       @if
;               some statements..
;       @endif
; statements between @if and @endif, executed if ax==1 and bx==0, or
; if ax==0 and bx==1.
; $xXX stack is used for condition satisfied entry label.

@mc     macro
        $xlist
        $setn   $x%$stack,%$num         ;; set identical number into $xXX stack
        $inc    $num                    ;; and advance counter
        $push   MC_PTR                  ;; push ID value
        $list
        endm

@cond   macro   cond,range
        $xlist
        $check  MC_PTR                  ;; first, check ID value
        ifb     <range>                 ;; check long range jump required.
                $jncond cond,$l%$val    ;; short range, jump on falure
        else
                $jcond  cond,$+5        ;; long range, jump on success
                $jcond  mp,$l%$val      ;; long range jump to falure point.
        endif
        $list
        endm

@or     macro
        $xlist
        $check  MC_PTR                  ;; first, ID cheking
        $setn   $val,$x%($stack-1)      ;; get top of $xXX stack
        $jcond  mp,$l%$val              ;; non conditional jump to success point
        $pop    MC_PTR                  ;; update top of $nXX stack
        $label  $l%$val                 ;; create current falure point
        $push   MC_PTR                  ;; then continue following evaluation
        $list
        endm

;----------------------------
;条件分岐構造
; if-else-endif structure
; standerd coditional branch.
; if argment condition is satisfied, statements between
; @if and @else are executed. not satisfied, between @else
; and @endif executed.
; if @else is omitted, statements between @if and @endif are
; executed only when condition satisfied.
; multiple condition definition is useable with @if macro.
; in the case, make condition arugment as blank.
; @e_if, special case of @if can be used in the structure.
; this may appear just after @else.
; statements between @else and @e_if are executed, and condition
; prepaired. then, it tested with @e_if. @e_if acts just as @if
; exepts its position. @e_if is used for multiple branch.
; following example
;       cmp     ax,1
;       @if     z
;               statements 1
;       @else
;       cmp     ax,2
;       @e_if   z
;               statements 2
;       @else
;               statements 3
;       @endif
; shows the usage of @e_if.

@if     macro   cond,range
        $xlist
        $inc    $num                    ;; reserve a number for label of end of structure
        ifb     <cond>                  ;; multiple condition check
                $pop    MC_PTR
                $jcond  mp,$+5          ;; comming here, means passing all condition check
                $label  $l%$val         ;; failure entry point definition
                $jcond  mp,$l%$num      ;; skip to @else or @endif
                $setn   $val,$x%$stack  ;; and define success entry point
                $label  $l%$val         ;; for @or statement
        elseifb <range>
                $jncond cond,$l%$num    ;; short jump, on reverse condition
        else
                $jcond  cond,$+5        ;; long range condition jump
                $jcond  mp,$l%$num
        endif
        $setn   $x%$stack,%($num-1)     ;; set special value for $xXX stack
        $push   IF_PTR                  ;; and push ID
        $list
        endm
```

ワンラインNEWS
韓国最大のエレクトロニクスメーカ，三星電機は米カリフォルニア州のシリコンバレーに超小型パソコンの研究開発拠点「US・PC・R&Dセンター」を設立．

の１という値は$n2に保存されているので，@elseや@endifは$n2（もちろん，$stackの値に依存）の内容を取得することで適切なラベルを生成することができるのです．

$popや$checkを呼出すと，このラベル生成のための値が$valに設定されるので，これと$labelマクロを使用すればラベル生成が容易に行えます．

$x??/$n??

このほかに，@ifや@condでは$x??というスタックも使用しています．@e_ifなどは本来のブロック構造を破壊するのです．ですから，きちんとしたスタック管理だけでは実現することができません．ひとつの「統一した」飛び先を記憶しておく必要があるのです．$x??スタックは，そのために使用されます．これは@ifや@condでのみ定義され，ブロック内部で変更されることはありません．@elseでは，$x??を参照することで，単一のブロック出口へのジャンプを生成しています．

同様な目的で使用される$n??スタックは，ブロック内部に@orや@elseが出現すると更新されてしまいます．これらに更新されないような値を保存するために，$x??スタックを使用しています．

繰り返し系のマクロでは，あらかじめ$numの値をインクリメントした後$pushを呼出します．このときの$numの値からループ先頭を示すラベルが生成されるわけですが，スキップされた値はループ終端を示すラベルのために使用されます．このループ終端を示すラベルは，@breakで使用されることになります．

@whileでは，ブロックIDを付け変えているのですが，ブロック自体が変更されているわけではないので$n??

```
@else   macro
        $xlist
        $setn   $val,$x%($stack-1)      ;; get special stack value
        $jcond  mp,$1%$val              ;; and jump to end of structure
        $pop    IF_PTR                  ;; create new $nXX value
        $label  $1%$val                 ;; and define 'falure' entry for @if
        $push   IF_PTR
        $list
        endm

@endif  macro
        $xlist
        $pop    IF_PTR                  ;; first, create normal falure entry
        $label  $1%$val
        $setn   $val,$x%$stack          ;; then special entry shows end of block
        $label  $1%$val
        $list
        endm

@e_if   macro   cond,range
        $xlist
        $pop    IF_PTR                  ;; similar to @if, but updating
        ifb     <range>                 ;; stack value
                $jncond cond,$1%$num
        else
                $jcond  cond,$+5
                $jcond  mp,$1%$num
        endif
        $push   IF_PTR
        $list
        endm

;-----------------------------
;繰り返し構造
; begin groups
; there is 3 loops with @begin.
; @begin - @again loop. this loop repeated forever.
; to exit this loop, use @break statement.
; @begin - @until loop is a conditional loop. if condition
; satisfied at the end of loop, it terminated.
; @begin - @while - @repeat is more complicated. first, statements
; between @begin and @while are executed. if condition checking fails
; at @while, loop terminated, and execution is skipped just after @repeat.
; if it succeeds, statements between @while and @repeat executed, and
; execution restarts just after @begin.

@begin  macro
        $xlist
        $inc    $num                    ;; reserve a number shows loop end
        $label  $1%$num                 ;; create loop top label
        $push   BEGIN_PTR               ;; and set ID
        $list
        endm


@again  macro
        $xlist
        $pop    BEGIN_PTR               ;; check ID
        $jcond  mp,$1%$val              ;; jump to loop top
        $label  $1%($val-1)             ;; and define loop end label
        $list
        endm

@until  macro   cond
        $xlist
        $pop    BEGIN_PTR               ;; check ID
        $set    $temp,$1%$val           ;; get loop top label
        if      ($+2-$temp) gt 127      ;; and calc jump range
                $jcond  cond,$+5        ;; long range
                $jcond  mp,$1%$val
        else
                $jncond cond,$1%$val    ;; short range, direct jump
        endif
        $label  $1%($val-1)             ;; and define loop end
        $list
        endm

@while  macro   cond,range
        $xlist
        $swap   BEGIN_PTR,WHILE_PTR     ;; modify ID
        ifb     <range>
                $jncond cond,$1%($val-1)        ;; short range exit
        else
                $jcond  cond,$+5        ;; long range exit
                $jcond  mp,$1%($val-1)
        endif
        $list
        endm

@repeat macro
        $xlist
        $pop    WHILE_PTR               ;; ID checking
```

の値を変えられないことから，専用の$swapマクロを使用しています．

以上のような概要とプログラム中のコメントから，ある程度動作は把握していただけると思います．このライブラリ自体は作成してから日が浅いため，まだまだ改良しなければいけない点が残っています．

たとえば，多重条件のネストや@untilなどでの多重条件のサポートなどです．しかし，あまりプログラムを長くし過ぎると紙面を圧迫してしまいますので，そのあたりは読者への課題，ということにしておきましょう．

いずれにしろ，かなり柔軟な記述がMASMのマクロで可能になる，ということがおわかりいただけたと思います．ここでは使用しませんでしたが，MASM5.1/TASM1.0には文字列の分解・結合をサポートする疑似命令が存在します．それらを使用すると，さらに複雑な処理もできるのです．ポテンシャルは非常に高いのですから，きちんと管理してやれば，マクロによる有用なライブラリを作成することは，むずかしいことではないのです．

さて，次号ではマクロ定義マクロを使用した例や，構造化マクロを利用したプログラム例などを紹介することにします．

```
        $jcond  mp,$1%$val              ;; jump to loop top
        $label  $1%($val-1)             ;; and define loop tail
        $list
        endm

;---------------------------
;ループからの脱出
; break/continue
; these macro are used in the loop.
; break: immidiately exit loop execution
; continue: immidiately jump to the top of loop

$search_loop    macro                   ;; this macro finds top level loop
        $setn   $val,$stack-1           ;; stack pointer works
        rept    $stack
                $set    $temp,$i%$val   ;; check $iXX stack value
                if      ($temp ne IF_PTR) and ($temp ne MC_PTR)
                        exitm           ;; loop ID, exit
                endif
                $dec    $val            ;; otherwise, check next level
        endm
        if      $val lt 0               ;; not loop ID found
                .err                    ;; cause error
                $if2
                        %out    COND Error: break/continue from outside of loop
                $endif2
                exitm
        endif

        $set    $temp,$n%$val
        $setn   $val,%$temp
        endm

@break  macro
        $xlist
        $search_loop
        $jcond  mp,$1%($val-1)          ;; jump to end of loop
        $list
        endm

@continue       macro
        $xlist
        $search_loop
        $jcond  mp,$1%$val              ;; jump to top of loop
        $list
        endm

;---------------------------
;回数指定の繰り返し構造
; do-loop
; this loop is repeated given times.
; argument reg is used for loop counter.
; sv is loop counter initial value. ev is exitting value.
; if CX is reg, sv = 1, and ev = 3, loop executed 2 times,
; with CX = 1 and CX = 2.
; note: this structure is not so effective.

@do     macro   reg,ev,sv,range
        mov     reg,sv                  ;; loop initial value set
        $xlist
        $jcond  mp,$1%$num              ;; skip re-calculation of loop counter
        $setn   $num,$num+2             ;; save local label space
        $label  $1%$num                 ;; define loop repeat point
        $push   DO_PTR                  ;; set ID
        $list
        inc     reg                     ;; re-calculation of loop counter
        cmp     reg,ev                  ;; and exitting check
        $xlist
        ifb     <range>
                $jcond  ae,$1%($num-2)  ;; short range jump
        else
                $jcond  ae,$+5          ;; long ramge jump
                $jcond  mp,$1%($num-2)
        endif
        $label  $1%($num-3)             ;; define local label
        $list
        endm

@loop   macro
        $xlist
        $pop    DO_PTR                  ;; check ID
        $jcond  mp,$1%$val              ;; simply, jump to repeat point
        $label  $1%($val-1)             ;; define end of loop
        $list
        endm

        .list
```

Reintroduction to HARD WARES

プログラマのためのハードウェア再入門

鎌田正寿



ハードウェアといってもいろいろあります．コンピュータを構成するハードウェアに限定しても，筐体（基板とか電源とかが入っている箱のこと）もハードウェアですし，CPUとかLSIなどもハードウェアですね．プログラムを作ってそれを動かしているときには，あまり意識しないことが多く，意識せずとも素直にプログラムが書けて使えるハードウェアの環境がいちばんよいのです．だからといってプログラマがハードウェアのことを知らなくてもいいということにはなりません．

ここではあまりむつかしくない範囲で，プログラマが必要とするハードウェアのことについておさらいしてみましょう．

では，まずどのパーソナルコンピュータにもついているRS-232Cのことから…….

普通の用途なら9ピンで十分

RS-232Cは規格の名称で，シリアルインターフェースのことを指します．このRS-232Cには，信号レベルや信号の駆動方式などの違うRS-422Aや423，485などの仲間があります．RS-232Cよりも高いボーレートが可能となったり，より長いケーブルを使用することができたりします．これらは互いにケーブルさえ工夫すれば接続可能で，ソフト的に見るとほとんど違いはないとまず憶えてください．

RS-232Cに使われるコネクタは，D-SUBと呼ばれる形で25ピンのものが多く使われています．最近は，IBM PCに採用されているからでしょうか，9ピンのものも使われることがあります．9ピンのものを使うメリットは小型化です．もちろん25ピンのすべての信号は入りませんが，たいていの用途にはこれで十分なのです．

さて，25ピンの信号を見てみましょう．1番ピンというのがどのピンかはコネクタに小さく番号が振ってあるのでわかるはずです（写真1）．

ここには書いてありませんが，このほかにも外部クロック用の信号など，

写真1■25ピンのRS-232Cコネクタ

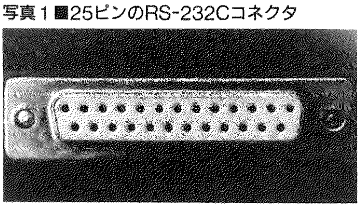


写真2■9ピンのRS-232Cコネクタ

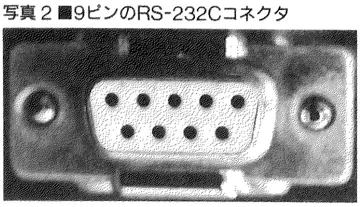


表1■RS-232Cコネクタのピンの機能

ピンNo.	方向	信号の名称		
1	−−	フレームグラウンド	FG	
2	出力	送信データ	TxD	(SD)
3	入力	受信データ	RxD	(RD)
4	出力	送信要求	RTS	(RS)
5	入力	送信可	CTS	(CS)
6	入力	データセットレディ	DSR	(DR)
7	−−	信号グラウンド	SG	
8	入力	データキャリア検出	DCD	(CD)
20	出力	データターミナルレディ	DTR	(ER)
22	入力	リングインジケータ	RI	(CI)

いくつかの信号が定義されています．それらは同期式などのためのもので，普通に使う分にはこの程度で十分です．また，このピン番号はターミナル，モデム，パーソナルコンピュータのほとんどの方式で使われているものです．

1番ピン（フレームグラウンド　FG）

フレームグラウンドは，フレームやシャーシに接続すべきグラウンドです．接続しなくてもほとんどの場合問題なく使えますが，ノイズのことを考えるとマジメに接続する必要があります．

2番ピン（送信データ　TxD）

送信データは，この信号を用いて1ビットずつ出力します．

3番ピン（受信データ　RxD）

受信データは，この信号を用いて1ビットずつ入力されてきます．

4番ピン（送信要求　RTS）

送信データがあるときにONとします．

5番ピン（送信可　CTS）

送信データを送ってよいということを示す入力です．CTS／RTSはハードフローコントロールに使われることもあり，そのときは，受信バッファがいっぱいになったときにRTSをOFFとして，送信側はそれをCTSで受けCTSがONのときだけ送信を行うようにします．ハードフローコントロールに使わないときには，RTSは常にON，CTSは無視するという使い方が多いようです．

6番ピン（データセットレディ　DSR）

送受信が可能であることを示すための入力です．ケーブルの先にターミナルが接続されている場合に，ターミナルの電源が入っているかどうかを判定するというような使い方をします．

7番ピン（信号グラウンド　SG）

FGをのぞくすべての信号のグラウンドです．非常に簡略化して接続する場合でも，このSGとTxD，RxDの3本だけは必ず接続しなくてはいけません．

8番ピン（データキャリア検出　DCD）

公衆回線などでモデムがキャリアを検出していることを示します．

モデムを使わない場合には，ソフト的に常にONとみなしたり，たんに無視したり，ケーブルでDSRと同じ信号を入力して使います．

20番ピン（データターミナルレディ　DTR）

6番ピンのDSRの逆で，こちらの機器が送受信可能であることを示します．

使い方は，こちらの電源が入っていることを単に示したり，モデムのコネクト状態を強制的に解除（つまり回線を切る）したり，モデムの自動着信を禁

止／許可したりするのに使われます．

22番ピン（リングインジケータ　RI）

モデムが接続されている場合に電話が鳴っていることを示します．この信号でコンピュータの電源をONにしたりするなどいろいろと応用は考えられますが，実際に使われている例は少ないようです．

なぜLSBから送るのだろう？

では，次に実際のデータの送受信方法を見てみましょう．データは，1ビットずつ1本の信号線を通じて送られます．まず，図1を参照してください．

横方向は時間軸で A -> B -> C……と進んでいきます．縦方向は，信号のレベルで(1)がONを，(0)がOFFを示します．データが何も送られていないときには(1)が続いています(A)．この状態から最初にスタートビットと呼ばれる1ビットの0がまず送られます(B)．このスタートビットによって，ひとつのキャラクタがどこからはじまっているかが認識されます．

続いてデータが送られます（C～J)．データはLSB（最下位ビット）が最初で，MSB（最上位ビット）が最後となります．この例では，データ長は8ビットですが，5，6，7ビットの場合もあります．

なぜMSBからではなく，LSBからなのかと疑問に思う方がいるかもしれませんね．これは私の想像なのですが，同じ‘A’というデータを7ビットで送る場合と8ビットで送る場合を考えてみてください．このとき，送る側と受ける側でデータのビット長が合っていれば問題はないのですが，一致していない場合にも，LSBから送ることにすればエラーにはなりますが，どのデータが送られているかは，データが7ビットで表現できる場合には判定できるはずです．これがMSBから送られると判定はできませんよね．

エラーになるにしても，なんらかの「手がかり」が得られたほうがいいと考えたのではないでしょうか（違うかなあ)．実際には，ハードウェアがどのデータのビット長でも必ずLSB側から送るために，簡単になるからという理由かもしれませんね．

ストップビットが存在する意味は？

さて，データの次はパリティビットが送られます（K)．パリティビットには，以下の種類があります．

①パリティビットは使わない

②偶数パリティを使用

（データ＋パリティの1のデータの数を偶数になるようにパリティビットを選ぶ）

③奇数パリティを使用

④常に1

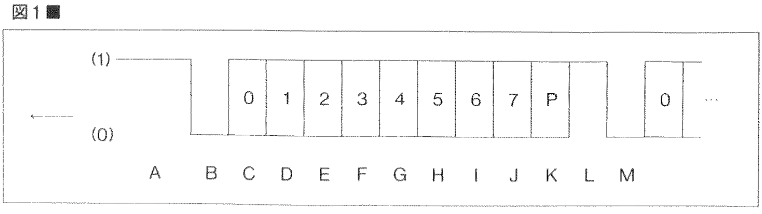
⑤常に0

日本国内では，8ビット目を使って漢字コードをそのままデータとして使う場合もあるので，データ8ビットでパリティは使用しないというのが多いようです．ASCIIだけを使う場合には，7ビットだけでデータは十分表現できますので，データ7ビットで偶数パリティが比較的多いようです．④と⑤の常に1とか0というのはあるにはあるようですが，実際に使われている例を私は見たことがありません．

パリティビットの次に最後に送られるのがストップビットです（L)．ストップビットは1ビット，1.5ビット，2ビットのいずれかです．歴史的には，もっと半端な1.25ビット（だったと思います）などもあるようですが，普通は1ビットか2ビットのいずれかです．このストップビットでキャラクタの終わりを示しますが，その存在にはいくつかの意味があります．ストップビットの直後には次のキャラクタのスタートビットがくるときがあります．データは1の状態と0の状態が（見た目には）あたかもランダムのように続きます．そのとき，「確かにキャラクタはここからここまでであり，今のところエラーはない」というような確認が必要になります．

スタートビットとストップビットがあることで，ボーレートやほかのパラメータが送る側と受け取る側で（確実にすべてではないにしろ）それほどまちがっていないことがわかります．なお，このストップビットが正しく1として認識できないエラーをフレーミングエラーと呼びます．

図1■



ビットの区切りをどうやって認識するのか？

ブレーク状態とかブレーク信号と呼ばれるものもこれらのデータで表現されます．表現方法は，最小のものはデータビットがすべて0のフレーミングエラーです（8ビットパリティなしなら最低10ビットの0が続く）．

原理的にはこれで認識できるのですが，ボーレートが不明の場合でも確実に相手に通知される必要があるので，通常0.2秒から0.5秒程度0の状態を続ける必要があります．これによって双方のボーレートが不明の場合でもブレークを正しく認識できます．

ブレーク信号は，キーボードからのCtrl＋Cと同じ働きをさせたり，ボーレートの調整に使ったり，モデムをコマンドモードに移行させたりなどに使います．

次にビットとビットの区切りをどうやって認識するかという問題があります．まず，送る側と受ける側でボーレートに同じ値を使うとします．ボーレートとは1秒あたりのビットレートのことを指します．このボーレートと同じタイミングでサンプリングを行っても送ることはできますが，正しく受信することはできません．

なぜなら，ちょうどスタートビットが開始されたところで最初のサンプリングをしたとしましょう．このとき，最後のストップビットまで送ったときと同じデータをサンプリングするためには双方のボーレートの誤差が0である必要があります．誤差が0であるというのは現実には望めませんので，結局は多くがエラーとなってしまいます．

ではどうしたらよいのでしょうか？そうですね，使うボーレートよりも何倍かの速さのクロックをもとにしてサンプリングすればよいのです．16分周の場合だと最初のストップビットが0となったのを検出したら最初のLSBを24クロック後に，以後16クロックごとにサンプリングすれば，ちょうどビットの真ん中を得ることができます．これらのことからボーレートの基準となるクロックの誤差は，片方のクロックが誤差0だとしても約5％以内．双方の誤差を考えて約2.5％以内．安全を考えると1％以下でないと安心して使えないことになります．

▶

どうでしたか？ふだんなにげなく使っているRS-232Cですが，改めて見てみるとおもしろいですね．

あまりピンの機能などを知らなくても不自由しないことが多いのですが，いざ標準的ではない組合わせで何かを接続しようとしたときなどは，このようなことを知っていると便利です．

COLUMN　ボーレートの自動認識はどうやって実現されている？

ATモデムは，最初に送られてくる文字を見てボーレートやデータフォーマットの自動認識をしています．これはどうやって実現しているのでしょうか．何人かに聞いてみても，知らない（または気にしたことのない）人が多いようなので，ここでその方法を推測してみましょう．

ATという文字が送られるのですから，ビット長が8ビット，パリティなしの場合には0100000101...0001010101となるはずです（最初の0はスタートビット，最後の1はストップビット，途中の...は不定長の1，データはLSBが最初であることを思い出してください）．

受信するデータのコードが決まっているのですから，最初のスタートビットで0になってからストップビットの直前に0になる場所も決まっています．この2つの0の間の時間がわかれば，それによってボーレートは計算できるはずです．しかし，通常RS-232CとはLSIを通じて接続されているだけで，そのLSIに直接ラインを監視したり，時間を計測できるような機能がついているようなことはまずありません．

そこで，選択可能なボーレートの内でもっとも速いものよりさらに何倍かで受信ボーレートを設定してしまうのです．このとき，もちろんエラーなしではデータは受信できませんが，エラーの有無にかかわらず，いくつのデータを受信できたかという数がボーレートと負の相関を持ちますので，それによってボーレートは判定可能となります．またAとTではパリティが異なりますので，パリティの判定もできるはずです．すべての組合わせをここで検討するには紙面がたりませんので，興味のある方は各自で検討してみてくださいね．

プログラム遊工房

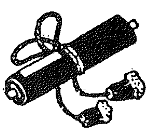
谷口典生

壱乃巻
テロップ表示プログラム

プログラム「遊」工房のコーナーでは，毎回，短くて楽しいプログラムを紹介していきます．
Cの勉強をはじめたけれど，文法を覚えるのにうんざりしている人，なにかプログラムを作ってみたいが何を作ったらいいかわからないという人は，とにかくリストを打込んで，動かしてみてください．説明を読んでプログラムの動きがわかったら，こんどは自分でいろいろと改造してみてください．きっと，プログラミングのおもしろさを発見してもらえると思います．
楽しく遊んで，いつのまにかちょっとしたテクニックが身についてしまう，そんなコーナーにしていきたいと思っています．

リスト1

```
 0: /*
 1:  * telop.c         画面にテロップを表示する
 2:  */
 3:
 4: #define WAITCOUNT       300    /* wait時間調整用定数 */
 5: #define TURBOC            0    /* Turbo C のときは 1 にする */
 6:
 7: #include <jctype.h>
 8:
 9: #if TURBOC
10: #define inp    inportb
11: #define outp       outportb
12: #endif
13:
14: unsigned char cback[] = "□";           /* バックグラウンド用キャラクタ */
15: unsigned char cforward[] = "■";        /* フォアグラウンド用キャラクタ */
16: unsigned int bcode[2];                     /* TVRAM 表示用に変換したコード */
17: unsigned char *str;        /* 表示文字列 */
18: int strpos=0;                       /* 表示文字列内の現在表示位置 */
19: unsigned int tmppat[16];            /* 表示している文字のドットパターン */
20: int tmpbit=0;                       /* 表示しているビット位置 */
21:
22: main(argc, argv)
23:         int argc;
24:         char *argv[];
25: {
26:         int inkey=0;
27:
28:         if(argc != 2) {         /* 引数が1個でなければ使い方を表示して終了 */
29:             printf("TELOP --- 指定された文字列をテロップ表示します。¥n");
30:             printf("usage : TELOP <文字列>¥n");
31:             exit(0);
32:         }
33:
34:         str = argv[1];
35:
36:         printf("¥x1b[>5h¥x1b[>1h¥x1b[2J");
37:         /* カーソル表示停止/最下行をプログラムで使用/画面クリア */
38:
39:         setbcode();
40:
41:         while(inkey != 0x0d) {
42:             updatescreen();
43:             wait();
44:             if (kbhit())
45:                     inkey = getch();
46:         }
47:
48:         printf("¥x1b[>5l¥x1b[>1l¥x1b[2J");
49:         /* カーソル表示開始/最下行をシステムで使用/画面クリア */
50: }
51:
52: setbcode()         /* ビット表示キャラの TVRAM 書き込み用コードを求める */
53: {
54:         unsigned int jcode;
55:
56:         jcode = jmstojis((cback[0]<<8) + cback[1]);
57:         bcode[0] = (jcode >> 8) + ((jcode & 0xff) << 8) - 0x20;
58:         jcode = jmstojis((cforward[0]<<8) + (cforward[1]));
59:         bcode[1] = (jcode >> 8) + ((jcode & 0xff) << 8) - 0x20;
60: }
61:
62: updatescreen()                                  /* 画面表示 */
63: {
64:         int i, j, base;
65:         unsigned far *p;                    /* TEXT VRAM へのポインタ */
66:
67:         p = (unsigned far *)0xa0000000L;    /* TEXT VRAM 先頭アドレス */
68:         base = 80 * 5;                      /* 表示先頭位置 */
69:
70:         tmpbit--;
71:         if(tmpbit < 0)
72:             getnextpat();           /* 次のパターンを得る */
73:
74:         for(i=0; i<16; i++, base+=80) {
75:             for(j=0; j<78; j++)
76:                     p[base+j] = p[base+j+2];
77:             p[base+j-1] = bcode[(tmppat[i]>>tmpbit)&1];
78:             p[base+j] = bcode[(tmppat[i]>>tmpbit)&1] + 0x8000;
79:         }
80: }
81:
82: unsigned int getkanjipat(jcode, pos)      /* 文字パターンを得る */
83:         int jcode;            /* JIS コード */
84:         int pos;              /* 行位置 (0-15) */
85: {
86:         unsigned int pat;
87:
88:         outp(0x68, 0x0b);     /* 漢字アクセスモードをビットマップにする */
89:         outp(0xa1, jcode & 0xff);
90:         outp(0xa3, (jcode>>8) - 0x20);
```



プログラムの使い方

画面を電光掲示板にする TELOP.EXE

リスト1を打込み，コンパイルして実行ファイルにしてください．このプログラムはPC-9800シリーズ用で，Turbo C, Quick CおよびMS-Cに対応しています(Turbo Cを使う場合は，5行目の「#define TURBOC 0」を「#define TURBOC 1」にしてください)．

実行するときの書式は，

TELOP 〈文字列〉

です．〈文字列〉には表示したい文字列を指定してください．文字列の中に空白が入る場合は，

TELOP "Programmer's Page 好評発売中!　"

のように" (ダブルクォーテーション)で囲ってください．文字列の最後まで表示すると最初の文字に戻って表示を続けるので，後ろにはいくつか空白をつけておくとよいでしょう．

好きな文字列を指定して起動すると，画面がクリアされ，右から左に大きな文字がゆっくりと流れていきます．眺めているといつまでも流れているので，終らせたいときにはリターンキーを押してください．

画面を流れる速度はリスト4行目で

定義している WAITCOUNT の値によって変えられます（値を大きくすると遅くなる）．機種やコンパイラによってかなり速度が違うので，自分の好きな速度に調整してください．

文字のドットを表示するために，黒四角，白四角のキャラクタが使われていますが，これはリストの14-15行目で設定しているので，バックグラウンドを空白にしてフォアグラウンドを黒丸にするなど，ほかのキャラクタに入れ替えて試してみるのもおもしろいでしょう．

プログラムの説明

❖main（）

telop.cのメイン関数

28-32行

引数の数が1個以外であれば，書式を表示して終了します．

34行

strには，第1引数（表示する文字列）へのポインタを代入します．

36，48行

「printf（"¥x1b[>5h¥x1b[>1h¥x1b[2J"）;」は，エスケープシーケンスを利用してカーソルの表示を停止し，ファンクションキーの表示をやめて画面をクリアしています．「¥x1b」は16進数で1Bhのコード（ESC）を意味していて，ESCに続く文字を指定することによって，いろいろな機能を実現できます．今回のプロ

```
 91:        outp(0xa5, pos);     /* 右半分のビットパターンを指定 */
 92:        pat = inp(0xa9);
 93:        outp(0xa5, pos+32);        /* 左半分のビットパターンを指定 */
 94:        pat += inp(0xa9) << 8;
 95:        outp(0x68, 0x0a);      /* 漢字アクセスモードをコードアクセスに戻す */
 96:        return(pat);
 97: }
 98:
 99: getnextpat()                /* 次に表示する文字のビットパターンを得る */
100: {
101:        int i, code;
102:
103:        code = getnextcode();
104:
105:        while((inp(0x60) & 0x20));
106:        while(!(inp(0x60) & 0x20));       /* VSYNC 信号が発生するまで待つ */
107:
108:        if(code < 0x2900 || code > 0x2bff) {
109:            for(i=0; i<16; i++)
110:                    tmppat[i] = getkanjipat(code, i);
111:            tmpbit = 15;
112:        } else {
113:            for(i=0; i<16; i++)
114:                    tmppat[i] = getkanjipat(code, i)>>8;
115:            tmpbit = 7;
116:        }
117: }
118:
119: getnextcode()          /* 次に表示する文字のコード(JISコード)を得る */
120: {
121:        unsigned char c1, c2;
122:
123:        c1 = str[strpos++];              /* 第1バイト */
124:        if(c1 == 0) {        /* 最後の文字を越えたら最初に戻る */
125:            strpos = 0;
126:            c1 = str[strpos++];
127:        }
128:
129:        if(!iskanji(c1)) {      /* 1byte文字の場合は2byte文字に変換する */
130:            c2 = hantozen(c1) & 0xff;
131:            c1 = hantozen(c1) >> 8;
132:        }
133:        else
134:            c2 = str[strpos++];   /* 第2バイト */
135:
136:        if(!iskanji(c1) || !iskanji2(c2)) {
137:            printf("¥x1b[>5l¥x1b[>1l");
138:            printf("Abort : 文字列の中に使えない文字があります¥n");
139:            exit(1);
140:        }
141:
142:        return(jmstojis((c1<<8) + c2));
143: }
144:
145: wait()             /* 時間待ち */
146: {
147:        int i, j;
148:
149:        for(i=0; i<99; i++)
150:            for(j=0; j<WAITCOUNT; j++);
151: }
152:
```

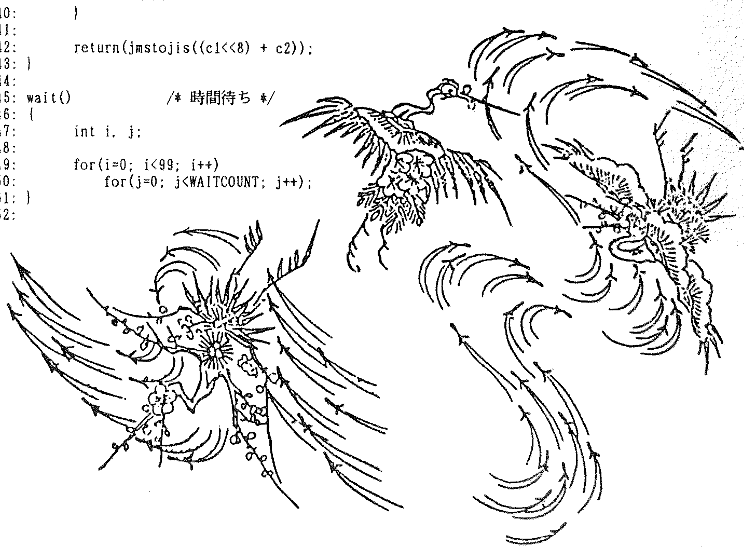


表1　TELOP.Cで使用したエスケープシーケンス

ESC[2J	画面クリア（「ESC＊」でも同じ）
ESC[>5h	カーソルの表示を OFF
ESC[>5l	カーソルの表示を ON
ESC[>1h	画面最下行をプログラムで使用
ESC[>1l	画面最下行をシステムで使用
ESC[y;x;H	y行xカラムにカーソルを移動

グラムで使っているエスケープシーケンスを表1にまとめておきます（エスケープシーケンスについては機会をあらためて詳しく説明することにします）.

41-46行

ここがこのプログラムのメインループになっています. updatescreen () でテロップを1列表示するごとに時間待ちwait () を呼び, もしなにかキーが押されていればそのキーコードをinkeyに入れます. inkeyの値が0Dhになるまで, つまりリターンキーが押されるまでループするようになっているわけです.「if (kbhit ()) inkey = getch () ;」という文では, キーが押されているかどうかをkbhit () 関数で調べ, 押されているときだけgetch () を実行するようになっています. getch () はキーが押されていない場合にはキーが押されるまで待つので, もしこの文を「inkey = getch () ;」という文に置き換えてしまうと, 1列表示するたびにキーが入力されるまで止まってしまうことになって具合が悪いのです.

◆setbcode ()

ビット表示キャラの TVRAM 書込み用コードを求める

56-59行

画面に出す文字列のドットを表示するキャラクタを, TEXT VRAM書込み用のコードに変換しています. TEXT VRAMに書き込むコードは, JISコードから2000hを引き, さらに上下バイトをひっくり返したものです. 通常, MS-DOSでは文字コードはshift-JISコードで扱われているので, ここでは関数jmstojis () によってshift-JISコードをJISコードに変換してから上下バイトを交換して20hを引いています.

◆updatescreen ()

TEXT VRAMへ直接画面表示

67行

TEXT VRAMに直接文字を書き込んで表示するため, farポインタにアドレスを代入しています. このテクニックについてはあとで詳しく説明します.

68行

表示先頭行へのベース値を設定しています.

70-72行

tmpbitは表示している文字の横のビット位置です. 1文字表示し終わると, 次の文字のビットパターンをtmppat [] に格納するためにgetnextpat () を呼びます.

74-79行

縦16ライン（ひとつの文字の大きさは「横16（または8）×縦16」）それぞれについて, 1列左にずらしてから右端に新しいビットを表示します. 各ラインのパターンtmppat [] の値をtmpbitの分だけ右シフトして最下位ビットをとることで目的の位置のビットを知ることができます.

◆getkanjipat ()

文字のビットパターンを得る

88-96行

JISコードと縦の行位置から文字のビットパターンを返します. I/Oポートにアクセスすることによって, 漢字ROMから文字パターンを得ることができます. GDCのモードコントロールによって漢字アクセスモードをビットマップに指定してから, 文字コードと行位置を書き込んでパターンを得ます（左右8ビットずつ2回に分けて読び出します）. 最後にGDCの漢字アクセスモードをコードアクセスに戻しておかないと, 画面に漢字が正常に表示されなくなるので注意が必要です.

◆getnextpat ()

次に表示する文字のビットパターンを得る

103行

getnextcode () をコールし, 次に表示する文字のJISコードを得ます.

105-106行

getkanjipat () を呼び出す前に, VSYNC信号の発生を待ちます. getkanjipat () はGDCの漢字アクセスモードを一時的に変更しているので, 帰線期間中に実行しないと画面表示が崩れてちらつきが出てしまうからです.

108-116行

縦の各行ごとに文字パターンをtmppat [] に格納します. JISコードが29xxh〜2Bxxhの範囲の漢字は「2バイト半角文字」と呼ばれているNEC特有の文字で, ドットサイズが「横8×縦16」と, ほかの文字の半分の幅しかないので, 特別に処理しています. tmpbitには文字の横幅（から1を引いたもの）を入れて返します.

◆getnextcode ()

次に表示する文字のコード（JISコード）を得る

123-127行

次に表示する文字のコードを得ます. 第1バイトをc1に代入しますが, c1が0であれば文字列を最後まで表示したことになるので, 文字列中の位置を示すstrposを最初に戻します.

129-134行

文字の第1バイトが漢字(2バイト文字)の第1バイトになっているかどうかをiskanji()で調べ，もし違えばhantozen()で漢字(2バイト文字)に変換します．漢字の第1バイトであれば，第2バイトをc2に代入します．

136-140行

不正な漢字が指定されたか，またはhantozen()で変換できない文字（1バイトカタカナなど）だった場合は，「使えない文字がある」と表示してプログラムを終了します．漢字ROMにはANK文字（1バイト文字）のパターンも入っているのですが，プログラムが少し複雑になるので，今回は漢字(2バイト文字)に変換できるものだけを扱うことにしました．

1バイト文字の英数字も画面上では幅が広くなってしまうので，幅をつめて表示させたい場合は「2バイト半角文字」を使ってください．

142行

shift-JISコードをJISコードに変換して返します．

wait ()

空のループを用いた時間待ち

149-150行

空のループによって時間待ちをします．プログラムの最初に定義しているWAITCOUNTの値を使っているので，文字が流れる速さが速すぎたり遅すぎたりする場合は，この値を変えて調整します．



ワンポイントテクニック

farポインタを利用したメモリ操作

画面表示の高速化

画面に文字を表示する場合，Cのプログラムではprintf()などの文字出力関数を使うのが普通ですが，TEXT VRAMに文字を直接書き込む方法もあります．この方法を使うと機種に依存したプログラムになってしまいますし，標準の関数が用意されているわけではないので，目的の位置に目的の文字を表示させるためにはアドレスとコードを計算しなければならないのが欠点です．しかし，通常の文字出力関数による表示に比べてはるかに高速な表示をすることができます．

今回のプログラムTELOP.Cでは，大きい文字をきれいに流すために高速な表示が必要だったので，TEXT VRAMへの直接書込みを使用しました．メモリへ直接アクセスするには，その部分だけアセンブラで書いてCのプログラムとリンクするという方法もありますが，ここではfarポインタを使って簡単にメモリへアクセスする方法を説明します．

farポインタの利用

Cのプログラムでは，文字列や配列変数を扱うときによくポインタを使います．TELOP.Cでも，リストの17行目で「unsigned char *str;」として表示文字列へのポインタを宣言しています．宣言をしただけではポインタが何を指しているのかわからないので，具体的にアドレスを与えてやる必要がありますが，ここでは，main()関数でstrにargv[1]の値を代入しています（34行目）．

8086系のCPUでは，アドレスをセグメントとオフセットによって指定します．セグメントとオフセットはどちらも16ビットで，0000h～FFFFhの値を取ります．Cのスモールモデルではデータエリアが64KB以内なので，ポインタのアドレスは16ビットのオフセットだけで決まってしまい，普通のポインタではデータエリア以外へアクセスすることはできません．ところが，「unsigned far *p;」などと「far」をつけて宣言すれば，セグメントとオフセットの値をもつfarポインタとなり，メモリ空間全体にアクセスすることが可能になります．

farポインタにアドレス，たとえばセグメントA000h，オフセット0000hを設定するには，「p = (unsigned far *) 0xa0000000L;」のようにします．long型の定数（0xa0000000L）に(unsigned far *)をキャストすることで，farポインタ型に型変換して代入しているわけです．逆にいえば，farポインタのアドレスはlong型に変換すれば上位16ビットがセグメント，下位16ビットがオフセットになっているということです．

アドレスさえ代入してしまえば，あとは普通のポインタと同じように使うことができます．ここでは，farポインタの指すデータがunsigned int型となるので，たとえばp[1]はセグメントA000h，オフセット0002hのワード値となりますが，「unsigned char far *p;」と宣言すれば，1バイト単位でアクセスすることができます．

リスト2

```
main()
{
        int i, j, x, y, d, l, c;

        for(c=0; c<20; c++) {
            d = 1;                                        /* 進む方向 */
            l = 1;                                        /* 進む長さ */
            x = 19;                                       /* 開始 X 座標 */
            y = 12;                                       /* 開始 Y 座標 */
            for(i=1; i<24; i++) {
                    for(j=x+l; x!=j; x+=d)
                            putxy(x, y, c&1);
                    for(j=y+l; y!=j; y+=d)
                            putxy(x, y, c&1);
                    d = d * (-1);                 /* 方向を反転 */
                    l = (abs(l)+1) * d;               /* 進む長さを増やす */
            }
        }
}

putxy(x, y, c)
        int x, y, c;
{
        printf("¥x1b[%d;%dH",y+1,x*2);  /* カーソルを y+1 行 x*2 列へ移動 */

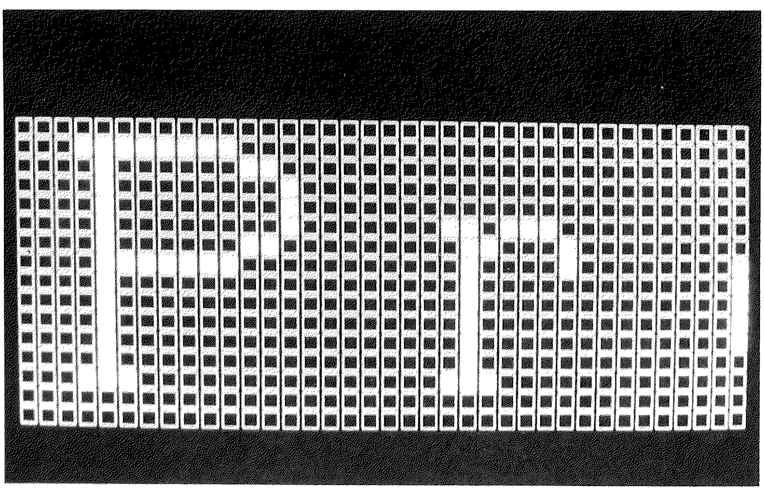
        if(!c)
            printf("●");        /* c が 0 なら黒丸を表示 */
        else
            printf("  ");        /* c が 0 でなければ消去 */
}
```

リスト3

```
putxy(x, y, c)
        int x, y, c;
{
        unsigned far *p;

        p = (unsigned far *)0xa0000000;           /* TEXT VRAM 先頭アドレス */

        p[y*80+x*2]   = c==0?0x7c01:0x20;
        p[y*80+x*2+1] = c==0?0x7c81:0x20;
}
```



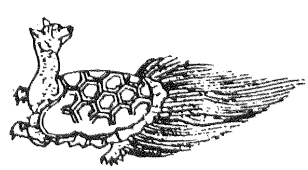
比較用プログラムの作成

それでは，実際に簡単なプログラムを作ってみましょう．まず，リスト2を打込んでください．画面中央から外側に向かって黒丸を回るように表示し，消すということを10回繰り返すプログラムです．

putxy () ではprintf () とエスケープシーケンスによって黒丸を表示しています．次に，putxy () をリスト3と入れ換えたプログラムを作ってみます．こちらは，farポインタによるTEXT VRAMへの直接書込みを行っています．

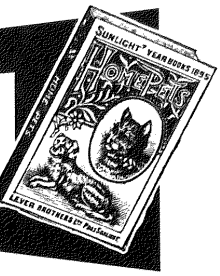
両方のプログラムを走らせてみてください．リスト2の方は黒丸が画面中心から外に向かって渦を巻くように描かれ，消えていくのがわかりますが，リスト3のputxy () を使ったほうは画面表示が速すぎて黒丸がどのように描かれているのかわからないほどです．TEXT VRAMへの直接書込みとprintf () による文字表示の速さの違いがよくわかると思います．

TELOP.Cのsetbcode () の説明のところで書いたように，TEXT VRAMに表示するために書く文字コードは，JISコードから2000hを引き，さらに上下バイトをひっくり返したものですが，全角の文字は画面上で半角文字2文字分なので，右半分にあたる部分には左半分に書いたデータに80hをたしたもの，つまりJISコードに6000hをたしてから上下バイトをひっくり返したものを書く必要があります（PC-9801VX以降の機種では右半分にあたる部分のデータを書き込まなくても漢字はきちんと表示されます）．





今月の受験情報



情報処理技術者になるメリットとは

金丸秀樹
株式会社ネットワールド

宅建試験に受験者が殺到

先日の新聞報道によると，宅地建物取引主任者資格試験，いわゆる宅建試験の受験者数が，過去最高だった昨年をさらに大きく上回っているとのこと．なかには，受験申込みの窓口に「待ち時間2時間」の看板が立つほどの長蛇の列ができたところもあるようです．

宅建資格は，土地や家屋・マンションといった不動産の取引業務に従事する者に取得が義務づけられているものです．最近の金利上昇や新土地税制実施の動きも何のそのといった勢いの不動産ブームを反映して，とくにここ数年受験者数はウナギ登りのフィーバーぶりです．いったい何がそんなに多くの人をこの試験に駆り立てているのでしょうか．

現在の宅建試験のひとつの特徴としては，女性や学生の受験者がめだって増えてきているということがあげられます．ブームに刺激されて宅建資格に興味をもった主婦や，「就職に有利だから」と受験する大学生が増えているのです．「不動産は儲かるから知っておいて損はないはず」とか，「カルチャーセンターに通っても何も残らない．すぐに活用しなくても資格だけは取っておきたい」といった動機も聞かれます．

しかし，この大幅な宅建受験ブームを作り上げている最大の要因は，やはりなんといっても不動産投機熱に乗りおくれずに業容拡大をはかりたいという企業の論理でしょう．不動産会社や金融機関が，宅建資格をもった社員をより多く確保したいと社内制度をもうけて受験を義務づけたり，信託銀行に早々と就職が内定した学生が銀行側から「できれば入行前に資格を取ってほしい」といわれるという状況があります．一方では，「就職に有利なだけでなく，入社後のことも考慮して，在学中に資格を取ってしまおう」という不動産会社志望の女子学生などがいます．企業サイドの価値観と「企業側が求めている資格を先取りしておいたほうが有利」という目先の利いた受験者の判断もあり，その相乗効果によってフィーバーともいえる状況を作り上げているといえるでしょう．

合格した方が得……なのは当たり前

情報処理技術者試験の場合を見ても，上の宅建試験のケースと同様に，あいかわらずの「コンピュータ時代」「情報化社会」，そして「プログラマが大幅に不足する時代の到来」と騒がれている状況の中で，受験者は年々増加の一途をたどっています．しかし，情報処理技術者試験の場合には，宅建やその他最近脚光を浴びているいくつかの資格試験とはひとつ大きな違いがあるのです．それは，この情報処理技術者試験に合格することが，情報処理を職業とすることの必要条件とは定められていない点です．たしかに，試験案内書の中にも次のように明記されているのです．

> 他の国家試験と異なり，合格者に対して資格・免許などを付与するものではありません．すなわち，合格者でなければ情報処理の業務に従事できないといったものではありません．

それでは，情報処理技術者試験に合格するということは，たんにその人の名誉だということにとどまってしまうのでしょうか．それはあまりにも美しすぎます．

「試験」というものが，そもそも合格することによって何らかの地位や資格を獲得するためのゲームとして存在し，結局は合格者本人に利益をもたらすチャンスである以上，情報処理技術者試験に合格するメリットというの

ワンラインNEWS
キヤノン販売は米国カリフォルニア州に輸入拠点を設置する．現地に拠点をおくことで新しい製品を発掘し輸入部門売り上げ比率を10%に高める方針．

も，さきに紹介されているような本音と同様に「自分にとって有利だから，少なくとも損はない」ということになるのでしょう．さらに，システムエンジニア，プログラマといった情報処理技術者がこれほど求められている時代なのですから，そのような「社会の要求」に合致した勲章をもっていれば，その価値は高い，つまり「少なくとも損はない」どころか「確実に自分に利益をもたらす」という判断が働くのも当然といえましょう．業績を拡大し利益を伸ばすために，なるべく多くの合格者を確保したいという企業サイドの逼迫したニーズと，そうした時代の要請に乗って自分の地位を確保し利益を得たいという個人レベルの欲求といった構図が，資格・免許の有無に関わらず情報処理技術者試験や宅建試験を過熱化させているのは確かでしょう．

情報処理技術者試験とは

情報処理技術者試験は，1969年度に通産省が情報処理技術者の育成とレベルアップをめざして発足させた認定試験です．その後は図1のような歩みをたどって現在にいたっています．試験の目的は，「情報処理の促進に関する法律」で次のように触れられています．

①情報処理技術者に対して目標を示し，刺激を与えることによって，その技術の向上を図ること．
②情報処理技術者として備えるべき能力についての水準を示すことにより，教育水準の確保に資すること．
③情報処理技術者の評価に関して，客観的な尺度を提供すること．
④広く受験の機会を提供することにより，国民各層の情報化に対する意識をかん養すること．

また，その性格については，医師や弁護士などのような独占資格や，事業者に有資格者の確保を義務づける必要資格とは異なり，技術の向上などを図るための推奨的な制度であるとしています．そのため，受験資格も「システム監査」と「特種」に年齢制限があるだけで，そのほかの付帯制限はいっさいつけられていません．

企業にとってわかりやすい試験区分

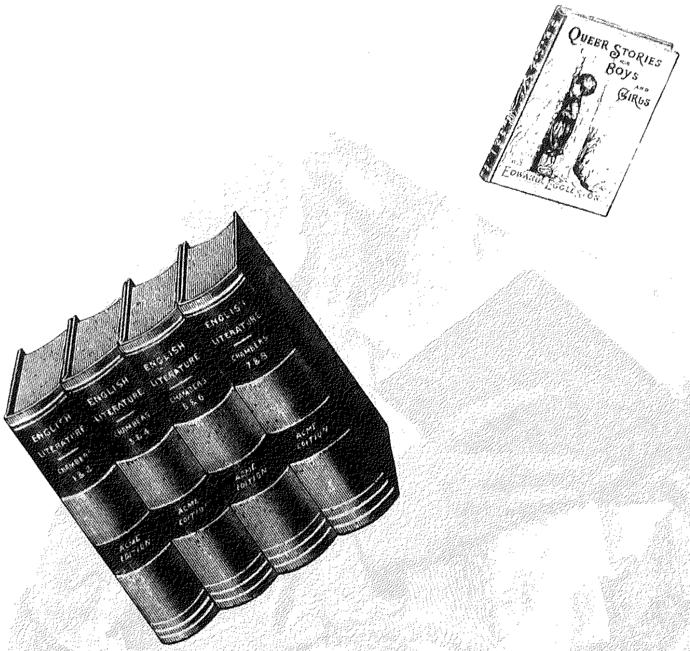
情報処理技術者試験で現在設けられている試験区分は，図2のようになっています．この中で，「システム監査」は1986年度から，また「オンライン技術者」は1988年度から加えられました．これは，情報処理業界の範疇が拡大し，ひと口に情報処理技術者と呼ばれる人々のカバーすべき範囲がどんどん広がっていく動きに対応してのものととらえることができます．最近の傾向としてソフトウェア関連企業の方向性が，いわゆるシステムインテグレータ的な統合化指向と，差別化を狙った専門特化指向との，二極分化をたどりつつある中で，今後とも新しい試験区分が設けられていく可能性は高いでしょう．

また，現実の企業活動の現場では中心となる情報処理技術者が，システムエンジニア，シニアプログラマ，プログラマなどと呼ばれ，職種が設けられていますが，それらの職種に求められる業務範囲との対応を意識して試験区分が設定されているのも情報処理技術者試験の特徴です．この試験の旧案内書の中には，図3のような表現で試験水準の解説がなされています．それによれば，特種情報処理技術者はシステムエンジニアとして，また第1種情報処理技術者はシニアプログラマあるいはアシスタントシステムエンジニアとして，そして第2種情報処理技術者はいわゆるプログラマとして，それぞれの役割期待が全社的なシステム開発の流れに沿ってすぐわかるように対応づけられています．

このような対応関係がはっきりされているということは，当然受験する者にとっても「自分がどこにいるか」を把握しやすいと思いますが，そのメリットは，企業サイドにとってプログラマ，シニアプログラマ，システムエン

図1　情報処理技術者試験の沿革

1969年	「1種・2種」第1回試験 (通産省の施策により情報処理技術者認定試験制度として発足)
1970年	「1種・2種」第2回試験 (情報処理振興事業協会などに関する法律の制定・施行により同法律第6条に基づく国家認定試験となる．これにより名称も「情報処理技術者試験」となる)
1971年	「特種」第1回試験 試験事務を通産省から（財）日本情報処理開発協会「情報処理試験センター」に委譲
1986年	「システム監査」第1回試験 (この年，前出法律の名称が「情報処理の促進に関する法律」に変わる) 「2種」のみ4月と10月の年2回実施となる
1988年	「オンライン技術者」第1回試験 試験実施時期の見直しが行われる (1種の実施時期を10月から4月に変更，ほかは変更なし．1989年4月試験から実施



ジニアなどといった一見あいまいとした技能職種を，組織の中でポジショニングしていく上で明確な判断材料としやすいということでしょう．さらに，「今はとにかく，より高いレベルの情報処理技術者をより多く確保したい．人材さえいれば業績が伸びる時代だ」とか，「(その企業内の情報処理技術者試験認定取得者数が評価項目のひとつとなる）通産省のシステムインテグレーション企業認定を取得し，各種の優遇措置や社会的な認知を得て業績拡大を図りたい」といった企業サイドの本音にもっとも近い試験制度であるともいえましょう．

受験者としてのメリットは

そのような状況を背景として情報処理関連企業は，宅建資格の例と同じように，情報処理技術者認定を採用の基準としたり，あるいは社員に情報処理技術者試験を受験・合格させようといろいろと特典を制度化したりしています．私企業の社員である以上，基本的な査定は実際の業務評価でなされるものでしょうが，次のようないくつかのかたちで合格者に対して特典を与え受験を奨励している企業がほとんどのようです．

①合格一時金を支給する

第2種合格者2万円，第1種合格者3万円，特種合格者5万円など，金額はまちまちですが，かなりの数の企業で実施されているようです．いわゆる情報処理関連企業でなくても，社員の自己啓発プログラムの一環として，情報処理技術者試験に対してもこのような制度を適用している企業は多いようです．また，受験手続きを企業側で一括して行ったり，合格した場合に限り受験料は会社負担にするなどが一般的です．

②資格手当として給与に上積みする

技能手当として，第2種1万円，第1種3万円，特種5万円などと付加給を与えている企業も多いようです．毎月継続して支給されるわけですから，基本給が同じであれば一時金よりも経済的なメリッ

図2　試験の対象および水準（情報処理技術者試験センター発行「試験案内書」より）

試験の区分	試験
情報処理システム監査技術者試験	情報処理システムの監査に主として，大学卒業程度の一般常識を有し，5年程度以上の一般実務を経験し，情報処理システムの企画・開発・運用および監査に関して，それぞれの専門分野についての知識を有し，システム監査を行い得る者を想定して試験を行う．
特種情報処理技術者試験	情報処理システムの分析，設計に主として従事する者を対象とし，大学卒業程度の一般常識を有し，3年以上の実務を経験し，それぞれの専門分野と電子計算機についての知識を有し，情報処理システムの分析と設計を行い得る者を想定して試験を行う．
オンライン情報処理技術者試験	ネットワークを利用する情報処理システムの分析，設計に主として従事する者を対象とし，大学卒業程度の一般常識を有し，3年以上の情報処理業務を経験し，情報処理ネットワークについての知識を有し，オンライン・システムの分析・設計・評価を行い得る者を想定して試験を行う．
第1種情報処理技術者試験	プログラムの設計，高度のプログラムの作成および第2種情報処理技術者などの指導に主として従事する者を対象とし，大学卒業程度の一般常識を有し，3年以上のプログラミング経験を有する者を想定して試験を行う．
第2種情報処理技術者試験	プログラム設計書に基づくプログラムの作成に主として従事する者を対象とし，高等学校卒業程度の一般常識を有し，1年以上のプログラミング経験を有する者を対象として試験を行う．

ワンラインNEWS
電子手帳用にいろいろなICカードが発売されているが，外出先で檀家の法事スケジュールがわかる「お寺さんカード」といった変わりダネもある．

トは高いといえます.

③昇進の条件とする

第1種に合格しないと主任になれないとか，特種でないと課長になれないなどの査定条件を人事規定に盛り込んでいる企業もあります．明文化されていなくとも，人事効果の際に合格者にはポイントを与えるように考課者に指導しているところもあるようです.

このような実状を見てみますと，情報処理技術者試験を受験しようとする者からすれば，たんに経済的なメリットだけに目をやらずとも,「取っておいて損はない」ということになりましょう．大がかりにいえば，時代が要請し企業が求めている認定資格なのですから，とくに第2種などは「とりあえず力試しに」と，時間を作ってチャレンジしてみましょう．第2種試験では，受験者の約40%を学生が占めています．彼らは，将来ソフトハウスや情報処理企業に入社するか，独立したプログラマになるか，あるいはまったく関係のない道に進むかは別として気軽にチャレンジし，そして合格者全体の約4分の1を占めています.

あなたがもしすでに情報処理関連の企業に勤務しているのなら，やがて，あるいはもうすでに，会社サイドから半ば強制的に「早く第2種に合格しろ」「次は第1種だ」とはっぱをかけられていることでしょう．まあ，それはやむをえません．時代の追い風を帆に受けて，自分の増収（この大量消費時代にわずかなものかも知れませんが）と，さらに高い責任ある職種へのステップアップという，自分自身のキャリアパスを築き上げて行くために，将来へ夢を追って飛び込んでみましょう.

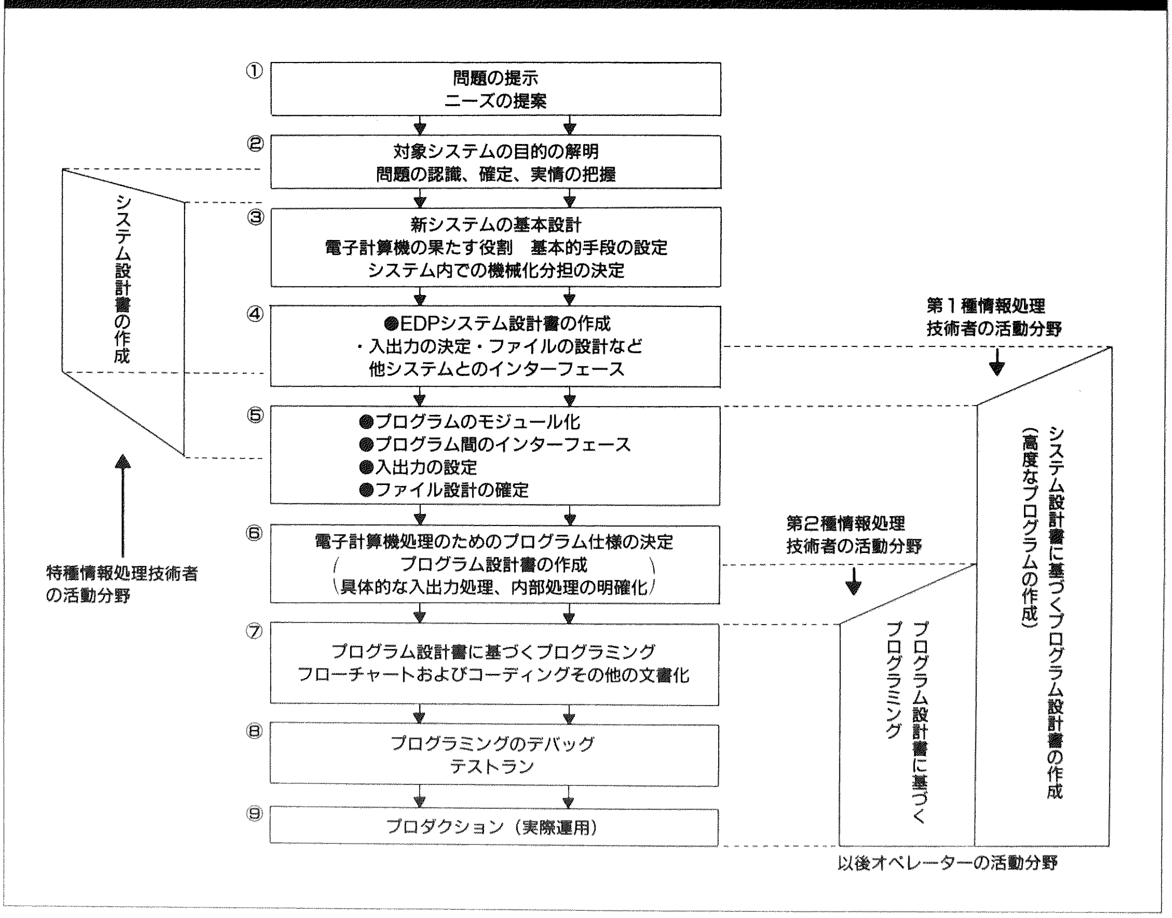
●情報処理技術者試験問い合わせ先

財団法人日本情報処理開発協会
情報処理技術者試験センター
〒105 東京都港区虎ノ門1-16-4
アーバン虎ノ門ビル8F
テレホンサービス：03-591-0429
事務局：03-591-0421

●試験願書受付と試験の月

1月：1種・2種願書受付
4月：1種・2種試験
7月：特殊・2種願書受付
10月：特殊・2種試験

図3　旧案内書での試験水準



小田嶋隆

オダジマタカシの

このページで，プログラミングについて連載を開始することになった．

今回は，初回ということで，お話だけにしておくが，次回からは，実際にプログラミングに取り組んで，その成果を報告する形で連載を進めて行くつもりでいる．

「同時進行ドキュメント」

というヤツだ．

よくある企画だ，と思う人もあるだろう．女優さんにパラグライダーをやらせたりするみたいな，安易にして新鮮味に欠けるレポーター番組じみたおちゃらけ企画だと思う人もいることだろう．

しかしながら，実際のところ，これ（セコいプログラムを組んで世間に恥をさらすこと）は安易どころか，非常に困難な作業だ．むしろ，壮挙と呼んでほしい．

だって，私は素人なのだから．

もう少し正確に言うと，私は，素人衆の間でこそパソコンの専門家ってことになっているものの，その実，専門家の間に入るとただの素人であったりする人間なのだ．

いや，というよりも，さらに正確を期して率直に申し上げるならば，私は，素人をたぶらかしてメシを食っている，いわゆる「半可通」なのである．

「音痴の歌手は，少なくとも音痴の音楽評論家よりは良心的だ」

といったのが誰だったのかは忘れたが（ボブ・ディランだったかもしれない），私は，その音痴の音楽評論家みたいなどうにもならない代物なのであって，スナックでカラオケのマイクを握らされことを恐れるあまり，いきなり泥酔することにしている困った先生なのである．

その，言ってみればコウモリ野郎であるところの私が，こういう専門家志向の雑誌の誌面でプログラムを披露するということは，何と言うのか，コウモリが鳥の前で模範飛行をしてみせることにも似た作業であるわけで，本人の立場から言わせてもらえば，こいつはやっぱりどうにもサマになりそうもない感じが，あらかじめしているわけなのだ．

ううむ．

まだ何も始まっていないのに，すでに私は愚痴を言いはじめている．

罪に先立つ悔恨

と，三島由紀夫は言ったが，こりゃさらにひどい．悔恨に先立つ弁解，失敗を予定した愚痴，撤退を前提とした行軍，終りの見えない始まり，始まらないままの終り，連載される中断，信仰告白を伴わないザンゲ…….

「反省してるか」

「はい」

「じゃあ，お前はどうして反省しなければならないようなことをやったんだ」

「いえ，あの，後で反省すれば良いと思って」

本題にはいろう．

1980年代のはじめ頃,「プログラミング」という言葉は，一種の神通力を持っていた．当時は，この言葉を口にするだけで，一部上場の企業に勤める課長クラスをピョンと飛び上がらせることができた．

結局，我々が住んでいるこの島国は，常に何らかの言葉が一時的な神通力を持っている長屋の井戸端みたいな場所なのだ．減塩，有酸素運動，善玉コレステロール，QC，自己啓発，マドンナ，おやじギャル，ワールドミュージック，ランバダ，ナイトクラビング，国際貢献，地球環境，カードライフ，ノーニュークス……と，我々は，常に新しい言葉を発明し続けている．そして，そうやって古い空虚な言葉を新しい空虚な言葉に言い替えながら，我々はいつもアタマの中味を新鮮に保とうとしている．しかし，それらの新しい言葉たちは，古いポリバケツに入った新しい生ゴミみたいに，はじめから腐っているのだ．

ともかく，あの黄金の（この言葉を「イエロー・マネー」と訳した中学生の頃，私は天才だった）黎明期においては，あらゆるパソコンメーカーがユーザーによるセルフプログラミングを推奨していたのである．

実際，

「誰にでも出来ます」
「ビジネスをベーシックで」
「家計管理も思いのまま」

とパソコンメーカーは言った．そして，人々はメーカーが嘘を言うはずがないと思っていた．

確かに，厳密にいえば，これらのセリフはまったくの嘘というわけではなかった．現実に自分のプログラムをビジネスに役立てた人もいたし，パソコンを利用して家計管理を自動化した人々もいた．しかし，その数は，日曜大工で家を建てた人よりさらに少なかったし，パソコンで家計を計算していた人々がそのプログラムの作成と管理に費やした時間と費用は，税理士を雇う場合よりもずっと高く付いたのだった．

要するに，醒めない夢が存在しない以上，「夢のマシン」たるパソコンに着せかけられていた夢もまた醒めざるを得なかったわけなのだ．

黎明期の「夢」を支えていたのは，プログラミングという作業によって展開される「無限の可能性」であった．そして，それは同時に「無限の不可能性」でもあったのである．

だから私は，生まれてはじめて書いたベーシックの入門書に

「プログラムは
あなた自身です」

と書いた．

なかなか,泣かせるセリフである．

が，このセリフをベーシックの入門書に書いた私は，結局のところ自身が組んだプログラム同様，支離滅裂な男であった．

ただいま

オダジマタカシのただいま悪戦苦闘中1

小田嶋隆

私が最初に書いたプログラムは

「駄じゃれプロセッサー」

という，およそ愚にもつかない代物だった．

それは，要するに入力された言葉の母音，子音を入れ換えて，よく似た音の新しい言葉を作るものだった．

と書いてみるとなかなか高度なプログラムであるようにも見えるが，決してそんなことはない．

なにより，わが「駄じゃれプロセッサー」は，入力も表示も半角のローマ字のみに限られていたし，母音，子音を入れ換えた上で表示される言葉は，ただの音の羅列に過ぎず，無意味な言葉ばかりだった．

たとえば「駄じゃれプロセッサー」に「karada」を入力すると，まずkaの音について，ka，ki，ku，ke，ko，と，母音を変えたパターンが表示され，次にa，ka，sa，ta，na，ha，ma，ya，ra，wa，ga，za，da，ba，pa……と，子音を変換したパターンが表示される．

で，以下，ra，daについても同様の手順が繰り返され，およそ60通りの「駄じゃれ候補群」が表示されるわけなのだが，こんなに煩雑な処理を繰り返したあげくに，ユーザーが読みづらいプリントアウトから発見できる駄じゃれは，「サラダ」だけだったのである．

日本語表示の問題を解決し，なおかつ，きちんとした辞書を備えて無意味語の排除が完璧にできるようにすれば，あるいは多少は役に立つ（それにしても，何の役に立つというのだろう）かもしれないが，私のプログラムははなはだ不徹底なもので，結局プリンター用紙のムダにしかならなかった．

それでも私は，作った時には，大いに得意になった．

「おい，オレは日本語の音韻要素の解析を実現したぞ」

などと吹聴して歩き，アスキーのコンテストに応募しようとさえした．

馬鹿な話だ．

結局，「駄じゃれプロセッサ」は，「民青の並木道」「人参なんてララーラララララーラ」「少年よ妻子を抱け」「天上天下唯我独身」といったような馬鹿な駄じゃれをいくつか生み出したのみで，PC-8001の製品寿命と運命をともにした．

次に書いたプログラムは，

「独裁者」

という名前のパーティー用ゲームで，これは自分で言うのもナンだが，なかなか良いデキだった．

説明しよう．

まず，プレイに先だって，ゲーム参加者の名前を全員入力する．

次に，「義務」を入力する．この「義務」というのは，要するにゲーム参加者がパソコンに命令された時にしなければならない行為のことで，内容，数には特に制限はない．たとえば「腕立て伏せ10回」「水をコップに3杯飲む」といった苦行の類でも良いし，「誰某から300円貰う」「右隣に座っている人にキスをする」「着ているものを一枚脱ぐ」ってな調子のものでもかまわない．

こうして，氏名，義務の登録が終ると，ゲームが始まる．

具体的にはパソコンがランダムに人名と義務を選択しながら10秒に1度，自動表示を続けるわけだ．

……と，たとえば水を10杯も飲んでしまったり，腹筋を100回やってスタミナ切れになったり，あるいは嫌な野郎にキスすることを命じられたりした奴から先に脱落して行く．これを続けて最後まで残った人間が勝利者ということになる．実に単純明快である．

このゲームを，たとえば高原のペンションあたりで，女の子を数名まじえてやったらさぞや楽しかったろうと思うのだが，残念なことに，私がこのプログラムを書いた当時，市場には行楽地やパーティー会場に持ち込めるようなポータブルなパソコンがなかった．それに，そもそも，一緒にパーティーを開くような女の在庫を私は持っていなかった（今でも同じだけどな）．

ともかく，「独裁者」は，全体でほんの10行ほどでしかない，INPUT文とRANDOM関数を使っただけの，ごく単純なプログラムだ．

この雑誌の読者の皆さんが，

嬌声飛び交うタイプのパーティー

と縁のある生活をしているのかどうかはわからないが，もし女子大生同伴のスキーツアーなどの機会に恵まれたら，ぜひお手持ちのブック型パソコンにこのゲームを忍ばせて，ぜひとも助平の役に立ててほしい．

青春は短い．そして，その，うたかたの夢のような，はかない青春の期間における唯一の鉄則は，どこまでも恥知らずであるべきだということだ．人生が一場の夢であり，青春が一瞬の旅であるのなら，旅の恥はかき捨て，どんな恥知らずになってでも助平の機会は逃すべきではない．

最初が愚痴で最後が説教かよ．

と思う人もあるかもしれないが，いま並べ立てた説教は，本当は愚痴だ．男というものは，三十代半ばを迎えて助平の機会がなくなってみてはじめて，自分が若い頃に助平機会をみすみす逃していたことに思い至るものなのだ．

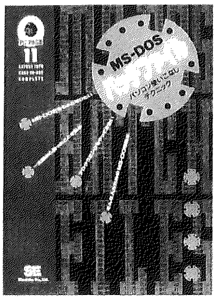
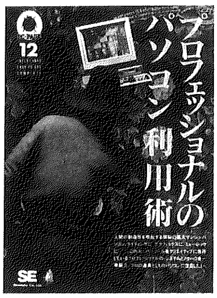
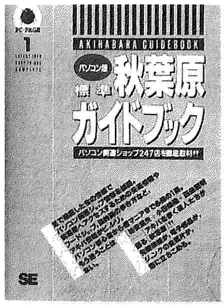
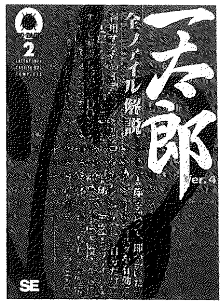
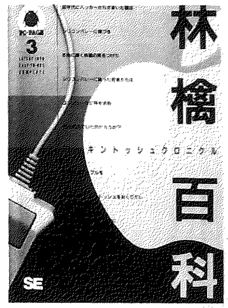
次回からは，実作のプログラムをネタに話を進める．

高度ではなくても，たとえば，読者諸兄の助平に貢献するようなプログラムが紹介できたら良いと思う．では．

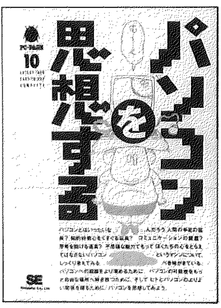
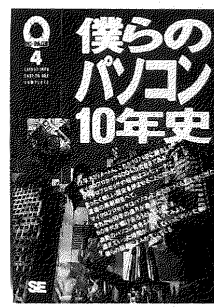
パソコン文化のトレンドを探る
バラエティーブックス
PC-PAGE
シリーズ

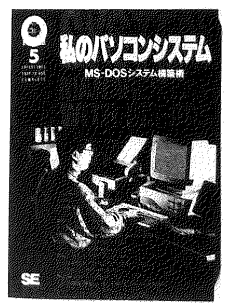
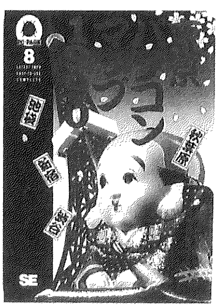
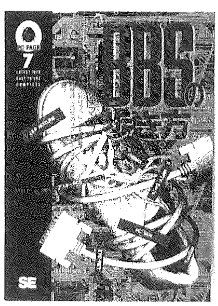
―毎月7日発売―

PC-PAGE

1．秋葉原ガイドブック ■ ¥1,200
2．一太郎Ver.4全ファイル解説 ■ ¥1,200
3．林檎百科～マッキントッシュクロクニル～ ■ ¥1,200
4．僕らのパソコン10年史 ■ ¥1,200
5．私のパソコンシステム ■ ¥1,200
6．パソコンブックガイド ■ ¥1,200
7．BBSの歩き方 ■ ¥1,200
8．東京パソコンマップ ■ ¥1,200
9．MS-DOS3,3全ファイル解説 ■ ¥1,200
10．パソコンを思想する ■ ¥1,200
11．MD-DOSバッチファイル ■ ¥1,500
12．プロフェッショナルのパソコン利用法 ■ ¥1,200

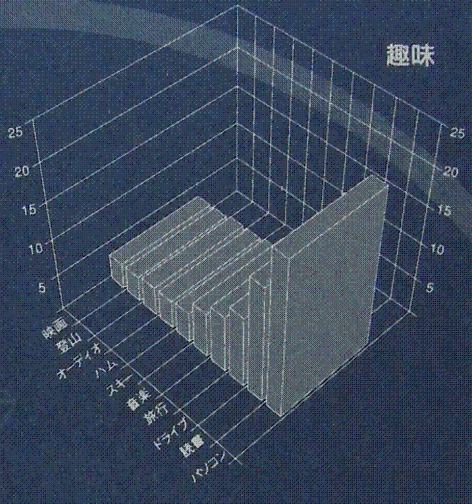





● PC-PAGE13「PDS白書」公開ソフトウェアをめぐる考察とレビュー 12月7日発売 ■定価¥1,500

株式会社 翔泳社 東京都千代田区平河町2-4-14 平河町KSビル
〒102 PHONE 03-263-0447 FAX 03-263-4526

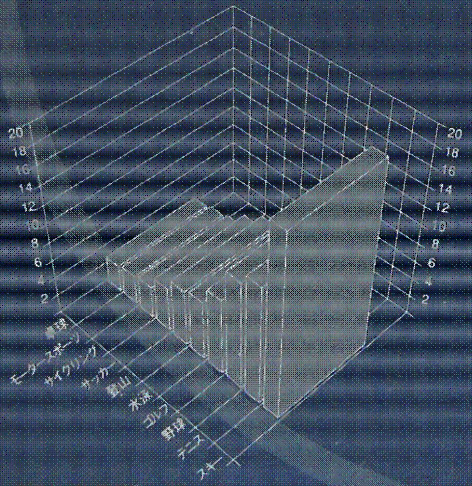
データ・ページ'91

このデータは，本誌創刊にあたって，プロ・アマチュアのプログラマを対象に実施したアンケート結果を集計したものです（有効回答者数：306名）．



趣味

information・page 2



好きなスポーツ

好きな音楽
またはミュージシャン

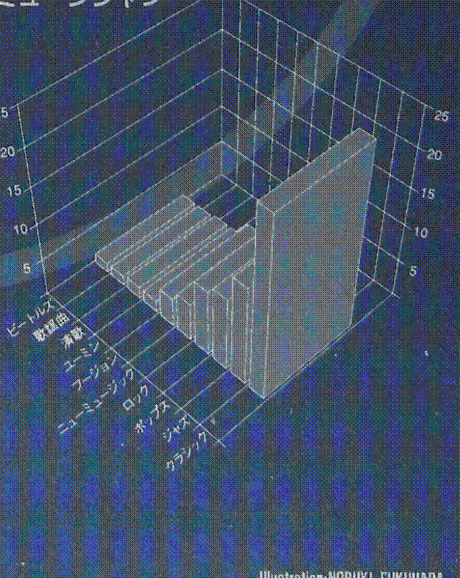


Illustration:NOBUKI FUKUHARA

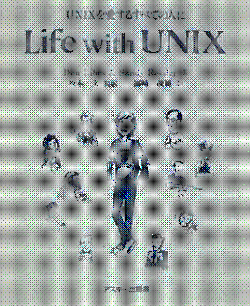
Book Review

自由なハッカー精神の健在を示す

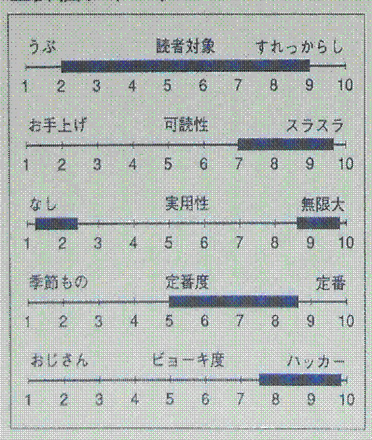
■中村正三郎

■Life with UNIX

Don Libes & Sandy Ressler 著
坂本文 監訳，福崎俊博 訳
アスキー出版局　418ページ　3000円



■評価チャート



毎月，この欄で書評を担当することになった．書評とは名ばかりで気ままな紹介である．よろしくお付き合い願いたい．

記念すべき第1回は，『Life with UNIX』．邦訳なのに原書と同じタイトルなのが面白い．

本書は Unix の雑学辞典だ．下らない話から資料性あふれる事柄まで貪欲に集め，ユーモアでくるんで品よくまとめている．読ませる工夫も随所にある．雑学だから話題は広く浅く，食い足りない部分がある．そのため的確な参考文献を掲げ，読者に配慮している．

原書に日本の事情を加筆しているが，これでは不十分だ．弁解は監訳者まえがきにあるが，中途半端な付け足しは却ってよくない．

大きく4部に分かれる．「1部　過去，現在，未来」では，Unix の歴史が語られる．ミーハー向けに Unix 界の有名人辞典も収められている．Unix のライセンス料の高騰，ソースへのアクセスの制限に触れているのは，著者，そして本書の性格を物語っている．ハッカー精神の賜物である．

「2部　情報」は，「印刷された情報」と「印刷されない情報」に分かれている．前者は出版物，雑誌などの紹介，後者は各種の会議，ユーザー会の紹介である．特に，書籍や雑誌の紹介コーナーは，本書の次を探す人には役に立つだろう．本欄にある読者対象や可読性を示すチャートは，本書の真似である．

「3部　UNIX の内部世界」は「ユーザーの環境」，「プログラマの環境」，「管理者の環境」に分かれる．「ユーザの環境」は，普通の意味での Unix 入門である．Korn shell やウィンドウの話が出てくるのは，いかにも最近の Unix 入門らしい．全くの初心者にはつらいだろう．参考文献がいる．

「プログラマの環境」は，Unix プログラマなら，これくらいは知っているし，また知っておいた方がいい知識をまとめている．

「管理者の環境」は，読んでもほとんど役に立たない．現実の管理者は，ずっと複雑でややこしい問題で毎日悩んでいるからだが，これから管理者になる人には少しは救いになる．パーソナル Unix マシンでは，泣いても笑ってもあなた自身が管理者だ．

「4部　UNIX の外部世界」には，アングラ情報と題した USENET や PDSの情報，コンサルティング会社や各種アプリケーションの紹介がある．特に「12章　現実に直面する UNIX」は，1部の未来の章と合わせて読むと楽しみが広がる．

1部と2部を読めば，Unix を1度も見たことも使ったこともなくても，いっぱしの Unix ユーザとして振る舞える．人名辞典を暗記し，かこみ記事の裏話を覚え，紹介してある本に全

て目を通せば，普通の Unix ハッカーなら頭を下げて通る，ペーパー Unix ハッカーになれる．つまり，ハイブロウな入門書，秘伝の奥義書，どう読んでもいい．

Unix は驚くべき広範囲のマシン上で標準 OS になった．この商業的成功とは裏腹に，我々は「Unix は死んだ」という言い方をする．企業間の政治の道具になったからだ．

Unix とそこで育まれた Unix 文化．これらを生み，育ててきた自由なハッカー精神は，もはや Unix の周囲から失われつつある．

本書は，かつての愛すべき Unix へのオマージュであると同時に，それを育てた精神が今なお健在であることを広く示し，Unix の死，ハッカー精神の死に歯止めをかけようとする試みであるとも言える．この点，品よくまとめすぎて悪意がないのが気にはなる．

目を通しておけば，パーティでの話題がぐっと広がる本である．

プログラマ向けの基本図書と90年の書籍動向

古田島義和（神田・書泉グランデ）

このコーナーでは，プログラマを対象とした和書・洋書の動向と定番本などを紹介していきたいと思います．今回は，まずこれからプログラマになろうとする方に必見ともいえる基本図書を紹介するとともに，90年全体のプログラマ向け図書の動きを見てみましょう．

もっとも基本的な図書として『ソフトウェア開発の神話』（企画センター）があります．さらに，『ソフトウェア考現学』（CQ出版），『プログラム設計の思想』（近代科学社），『コンピュータアーキテクチャ』（近代科学社），『オペレーティングシステムの概念／上・下』（培風館），『UNIXカーネルの設計』（共立出版／bit別冊），『アルゴリズムの設計と解析Ⅰ・Ⅱ』（サイエンス社），『コンパイラⅠ・Ⅱ』（サイエンス社）などがあり，プログラムやコンピュータの基本書としてあげられます．

そのなかで，『コンパイラⅠ・Ⅱ』は，発売以来快調な売れ行きを見せています．原書の『コンパイラ』（Aho著／Addison Wesley刊）のほうは訳本の出版以降も依然として洋書部門ベスト10に顔を出しています．コンパイラ関連では，『Lex & Yacc』（O'Reilly），『Compiler Design in C』（Prentice Hall）も人気を呼んでいます．和書では，『YaccによるCコンパイラプログラミング』（ソフトバンク）が売れています．

1990年を通してみると，X Windowではじまり暮れたという感じです．とくに，O'ReillyのX Windowシリーズ全8巻はベストセラーになり，『X Window programing & Application with Xt』（Prentice Hall／以下PH）は必読書として売れ，トッパンからの訳本もベストセラーになっています．PHは，第2弾としてMoti版を出版し，それが洋書ベスト10に入っています．Motifに関しては，PHから全5巻のシリーズが出版され，なかでも『Style Guide』『Programmer's Reference』『Programmer's Guide』が好調です．

次世代言語C++の和書が89年末頃から多くなっています．『An Introduction to Object-Oriented & C++』（Addison Wesley）の訳本をトッパンが出版してから次々と登場しています．最近では，『C++言語入門』（アスキー）が出ています．これは，PHの『Programing in C++』の訳で，内容がよくまとめられているだけに，しばらくは標準的な書籍となることでしょう．

UNIXネットワーク関連が，90年4月頃から動き始め，PHの『UNIX Network Programing』がロングセラーになっています．TCP/IPについては，『Internetworking with TCP/IP』（Comer著・PH）があり，共立出版の訳本とともに好評です．91年早々には洋書の第2巻が発売される予定です．

90年に話題になった出版物について書きましたが，91年はUNIX上のGUIの世界でUIのOpen LookとOSFのMotifがどういう方向になるのか，LANでのNetWareの動きなどが注目されるおもしろい年になりそうです．

■和書＆洋書ベスト10

（書泉グランデ調べ和書：1990年9月30日～10月27日洋書：1990年8月～9月）

1	健康ソフトハウス物語	ラディク／星雲社	1	Compilers (paper)	Addison Wesley
2	C言語を256倍使うための本	アスキー	2	UNIX Network Programing	Prentice Hall
3	VZを256倍使うための本	アスキー	3	X Window System Programming and Applications with Xt	Prestice Hall
4	X-Window OSF/Motifプログラミング	日刊工業新聞社	4	Computer Architecture A Quantitative Approach	Morgan Kaufmann
5	コンパイラⅠ	サイエンス社	5	Lex & Yacc	O'Reilly
6	プログラミング言語AWK	トッパン	6	Annotated C++ Reference manual	Addison Wesley
7	Cによるプログラミングスタイルブック	共立出版	7	OSF/Motif Sytle Guide	Prentice Hall
8	コンパイラⅡ	サイエンス社	8	OSF/Motif Programmer's Reference	Prentice Hall
9	入門NetWare	ソフトバンク	9	Art of Human-Computer Interface Design	Addison Wesley
10	プログラミング言語C　第2版	共立出版	10	Compiler Design in C	Prentice Hall

Programmer's Program

米国でトップシェアを誇る プログラマのためのエディタ 日本語Brief(ブリーフ)

同一パッケージで複数の国産マシンに対応

Briefは，日本ではあまりなじみのないエディタであるが，米国ではNo.1のシェアを誇っており，Microsoft C 6.0やQuick Cのインストール時にエディタのキーアサイン設定の選択で登場するほど有名である．このBriefを使えばプログラミング統合環境を簡単に構築することができる．

パッケージには，リファレンスマニュアルとマクロ言語ガイドが各1冊，クイックリファレンスが1部，登録カード，そして3.5インチのディスク2枚，5インチのディスク2枚の計4枚が入っている．Briefは，複数の機種に対応したエディタで，本体とハードウェアに依存するドライバ部分は独立しており，PC-9800シリーズ，J-3100シリーズ，PS/55シリーズ，FMRシリーズ(50/60/70/TOWNS)，AXなど主要な国産パソコンに対応している．さらに，パッケージに5インチと3.5インチの2DDのディスクが同梱されているので，ほとんどの機種でメディア変換を行うことなく利用できる．

インストールは使用機種を選択し，環境やディスプレイの設定を行うだけの簡単なものだ．Briefを起動すると，ウィンドウ枠と最下段のステータス表示だけのとてもシンプルな画面が表示される．この画面からはとてもBriefの実力は想像できない．

強力なプログラム支援機能

Briefの最大の特徴をあげるとすれば，プログラム支援機能であろう．使用する言語にあわせて，セットアップ時にコンパイル方法などの指定をすると，コンパイルや，エラー行の表示など一通りのことをひとつのキーで呼び出すことができ，もちろんソースを修正して行数が変化しても，次のエラー行へまちがいなくジャンプすることができる．また，複数の言語を指定しておけば，拡張子で自動的に呼出す処理系を判断してくれる．たとえばアセンブラとCのソースを同時に編集して，それぞれをアセンブル，コンパイルして，リンクすることが簡単にできる．

もうひとつの特徴は強力なマクロ言語である．Briefのマクロ言語はLispライクで，多くの強力なコマンドが用意されている．マクロプログラム中から自由に呼出して使うことができるが，最初はあまりのコマンドの多さにとまどってしまう．しかし，マクロ言語のリファレンスマニュアルは200ページ近くで構成され，用法だけにとどまらず，多くのサンプルも掲載しているので，すぐに理解できるであろう．

通常のインデントに加え，「{」や「}」にも対応した自動インデント機能や，テンプレート編集機能でC言語のif構文やwhile構文も生成してくれる．

また，ブロックの削除やパターンの置換などのほとんどの操作において，300回前までの取消しができる強力なアンドゥ機能，一定時間キー操作をしないとその間に編集結果を自動的に保存してくれるオートセーブ機能などは，ちょっとしたミスや事故から大切なソースファイルなどを守ってくれる．

ほかにも，マルチファイル・マルチウィンドウ機能，キーの再割当て機能，わかりやすいヘルプ・チュートリアル機能，正規表現をサポートした文字列検索・置換機能など，必要な機能は全て用意されている．

◆

Briefは，プログミングのために使うプロ好みのエディタといえる．米国での普及を考えると，今後は日本でもファンをつかむに違いない．

●日本語Brief（ブリーフ）
販売：日本システム株式会社
48,000円／実行環境／MS-DOS 3.1以上／メモリ384KB以上／ウィンドウ数55／同時編集ファイル数：制限なし／最大行数512桁

ワンラインNEWS
Macintoshはキヤノン販売のほかに日本電子計算（JIP）でも正規購入が可能だ．ただしJIPは企業や研究機関へのWSの販売を主としている会社である．

プログラマ登場①

ジャズ，空手，そしてプログラミング

株式会社日本オフィス機器
園田康博



園田康博（そのだやすひろ）
昭和23年福岡県に生まれる．東京理科大学理工学部経営工学科卒．空手道拳誠会師範．株式会社日本オフィス機器代表取締役社長．著書に『C言語活用研究』（共著　電波新聞社），『デジタル版徒然草』（翔泳社）などがある．

CP/M，MP/MからMS-DOSへ

1978年の会社設立当時，まだパソコンはなく仕事はミニコンあるいはオフコン用の業務ソフトの開発，使用言語はアセンブラのみだった．

開発用の端末としてPC-8001を購入したのがパソコンに関わるきっかけとなったが，実際にはRS-232Cの転送スピードが300bpsだったために端末としては使いものにならず，しかたなく内蔵されていたBASICでゲームばかりを作っていたとか．

パソコンが業務に関係するようになったのは，Z80上のCP/MまたはMP/Mの上でμCOSMOSを作成してからで，当時のシステムには，Z80マシンを中心にマルチタスクで6台端末を接続した大規模（？）なものもあった．PC-9801を使うようになってからも長い間CP/Mだったが，時の流れには逆らえず現在はMS-DOS上でMS-Cを使用している．

UNIX用ソフトの開発が中心

現在の仕事の中心はUNIXマシン上の業務ソフトの開発．最近は，日立のUNIXマシン用の株価情報システムなどを開発した．このときは，保存用にプリントアウトしたソースリストの山で事務所が埋めつくされ，人の入るスペースがなくなってしまったそうだ．

開発言語は100％がCで，事務所のPC-9801で作成したソースをクライアントのターゲットマシンに持ち込んでコンパイルとテストを行っている．デバッグは，すべて机上デバッグだ．

μCOSMOSの開発は，PC-9801RXとHXの2台で，エディタはMIFES，コンパイラはMS-C，アセンブラはMASM，特殊なツールはいっさい使わないというシンプルな環境で行われている．

島流しにされても大丈夫？

ジャズピアニスト，空手家，そしてエッセイストとしても知られる園田氏がコンピュータ関係に進学することを決めたのは，サイバネティックス関係の本を読んだことがきっかけだったという．しかし，大学でFORTRANによる「三角形の面積を求めるプログラム」を教わったときには，そのあまりのつまらなさに完全に道を誤ったと思ったそうだ．

そんな園田氏も，いまでは，たとえ島流しになってもコンピュータとプログラミングのマニュアルとモデムと電話線さえあればあとはなにもいらない，というほどコンピュータとプログラミングに惚れ込んでしまっている．

このところ，μCOSMOSの新バージョンのためのプログラミングの実験に夢中になっているというので，新しいμCOSMOSがどのような機能を身につけて登場するか，おおいに期待したいところだ．

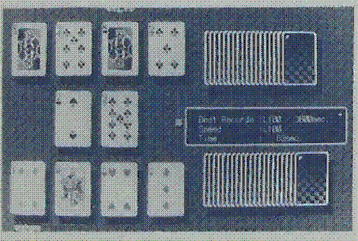
●μCOSMOS-MULTI＋

μCOSMOS-MULTI＋は，MS-DOSパソコン上で動作するリレーショナルデータベースソフトである．項目単位やレコード単位での実質的な文字数の制限がなく，長い文字列を含むデータの管理に向いている．また，外国語のフォントを自由に作成でき，ファイルサイズ無制限のワープロ機能を備えるなど，ほかのデータベースソフトには見られない多くの特徴をもっている．
製品名：μCOSMOS-MULTI＋
定価：88,000円
問い合わせ先：株式会社日本オフィス機器　TEL 03-582-1411

オンラインソフト採集帳 1

時間を競うトランプカードゲーム スピード

今回紹介するオンラインソフトは，トランプを使ったカードゲーム「スピード」．トランプゲームは，ポーカーやブラックジャック，そして7並べなどがすでにパソコンゲーム化されている．「スピード」がこれらのカードゲームと大きく違うのは，順番に自分の場が回ってくるのではないということと，競うのは点数ではなく時間ということだ．



2つの圧縮ファイルをダウンロードする

CRD. LZHとプログラムのバグ修正をしたCRD_FIX. LZHの2つのファイルをダウンロードする必要がある．ダウンロードしたら，LHARCを使ってCRD. LZH，CRD_FIX. LZHの順に解凍する．途中で「書き換えてもよろしいですか」と聞いてくるのでYと入力する．

解凍後のファイル数は15個．このうち拡張子が．PCKのファイルは画像データ，詳しいゲームの説明はMANUAL. DOCに，プログラムの説明はREADME. DOCに書いてある．SCREEN. COMは画面のグラフィックモードなどを設定するプログラム，ZUMI. COMは常駐型のマウスドライバだ．

本体はCRD1. EXEだが，マウスドライバの起動やスクリーンモードを設定する必要があるため，ゲームの起動はバッチファイルを使って行う．

```
A> CRD
```

と入力すればゲームが起動される．

ルールは簡単だが……

「スピード」は2人でプレイするゲームで，ジョーカーを除いた52枚のカードを使う．手前に4枚の場札と中央に1枚のマスターカードを，お互いに手札から開く．合図とともに自分の場札を調べ，中央に置かれた2枚のマスターカードのどちらかと連続する数字のカード（AとKは連続したカードとする）があれば，それをマスターカードに重ね，同時に常に場札が4枚になるように手札から1枚開く．どちらのマスターカードとも連続する数字のカードが場になくなったときは，再び合図とともに手札をマスターカードに重ねる．これを繰り返し全ての手札が早くなくなったほうが勝ちとなる．

単純で，はまりやすいから要注意

「スピード」では，対戦相手はコンピュータとなり，目的のカードをルールに従いマウスを使ってクリックするだけだ．1ゲームごとに，コンピュータは勝敗によって速度を上げたり，下げたりしてゲームを続行する．そして，

HEALTHY? ・ UNHEALTHY?

① 東京駅にはドリンク剤のバーがある



Illustration
MIKIO SASAKI

からだにいいのか悪いのかはともかく，ドリンク剤を飲みながら夜なべ仕事をするプログラマは少なくない（編集者も！）．「あれが効く」「いんや，これのほうがいい」という話を聞いていると，ドリンク剤の売れ筋が知りたくなってきた．調べてみるとなんとドリンク剤のバーがある．東京駅大丸の1階にある「エナジー・プール」だ．200種もあるというエナジー・プールのドリンク剤のベスト10を教えてもらった（表参照）．

いわゆるドリンク剤には，2つの系統がある．

ワンラインNEWS
韓国の大宇電子はソウル大と共同で3－6歳の幼児向け教育パソコンを開発．文字の読めない幼児でも操作のできる専用キーボードを備える．

最も速かった記録が保存される。

最初のうちはコンピュータが少し速くなると勝てなくなってしまうが，やめようとしない限り永遠にゲームは続く。負ければ悔しくて続け，勝てば気をよくして続ける，といった具合にいつまでたってもやめられない。ルールが単純なだけについついのめり込んでしまう習慣性の強いゲームだ。

カード画像を利用できるから……

ファイルの中には，カードの画像データとそれを利用するためのサンプルプログラムSUMPLE.C（ファイル名のまま）も付属している。カードのデザインはとても洗練されているので，このサンプルプログラムを参考にカードの画像データを利用すれば腕に自信のある人なら，カードゲームや占いなどを自分自身の手で作ることができるはずだ。

また，この画像データを使ったカードゲームなら本体をダウンロードするだけで手軽にカードゲームが楽しめることになる。

■スピード

分類：ゲーム／登録名：CRD.LZH CRD_FIX.LZH／著作権者：市川　泉（PC-VAN:MZG35394）／動作環境：PC-9800シリーズ／掲載月日：1990年8月15日／入手月日：1990年8月23日／入手場所：NIFTY-Serveソフトウェアギャラリー（GO FGAL）／アーカイバ：LHARC

ファイル構成

<CRD.LZH>

Name	Original	Packed	Ratio	Date	Time	Attr	Type	CRC
BAR.PCK	2336	507	21.7%	90-08-01	12:29:24	a--w	-lh1-	8885
BOTTOM.PCK	388	28	7.2%	90-07-29	0:46:24	a--w	-lh1-	F8E6
CRD.BAT	57	52	91.2%	90-08-15	16:40:30	a--w	-lh1-	DAA3
CRD.SCR	340	206	60.6%	90-08-15	16:16:58	a--w	-lh1-	9763
CRD1.EXE	87019	33579	38.6%	90-08-15	16:52:48	a--w	-lh1-	C0BD
LINE.PCK	676	170	25.1%	90-08-02	17:47:16	a--w	-lh1-	B37E
MANUAL.DOC	3106	1444	46.5%	90-08-15	17:55:54	a--w	-lh1-	8B99
MESSAGE.PCK	7219	332	4.6%	90-08-10	9:16:14	a--w	-lh1-	AD31
NULL.PCK	3028	215	7.1%	90-08-02	12:25:34	a--w	-lh1-	FA48
PARTS.PCK	122324	22884	18.7%	90-08-13	11:58:46	a--w	-lh1-	C572
README.DOC	2283	1203	52.7%	90-08-15	17:59:04	a--w	-lh1-	47FB
SCREEN.COM	10568	6976	66.0%	90-08-14	10:21:50	a--w	-lh1-	43D2
SUMPLE.C	1945	762	39.2%	90-08-15	17:23:34	a--w	-lh1-	49DD
TILE.PCK	388	111	28.6%	90-07-29	1:32:04	a--w	-lh1-	DEC3
ZUMI.COM	6036	1913	31.7%	90-08-15	10:15:52	a--w	-lh1-	4877
15 files	247713	70382	28.4%	90-08-15	12:06:16			

<CRD_FIX.LZH>

Name	Original	Packed	Ratio	Date	Time	Attr	Type	CRC
CRD1.EXE	87019	33589	38.6%	90-08-23	11:22:18	a--w	-lh1-	2B24
1 files	87019	33589	38.6%	90-09-03	14:49:30			

清涼飲料と医薬品の栄養ドリンクだ。当然医薬品のほうには「有効な成分」が含まれていることになるが，清涼飲料のほうもビタミンやミネラルを加えるなど工夫している。武田食品のタケダC1000などは近ごろヒットした清涼飲料だ。

女性向けのオリジナルドリンク剤「ローズウォーター」まで用意してOLへの売り込みを図っているだけあって，客層は男女半々とか。場所がらか，「ユンケル黄帝液」が中年男性の影を感じさせるだけで，フツーのビタミン剤的な栄養ドリンクが多い。プログラマが得意とする「目醒まし用のドリンク剤」は姿を見せていない。

★エナジー・プールドリンク剤ベスト10

★1．リゲイン（三共）300円
★2．リポビタンD（大正製薬）130円
★3．アリナミンV（武田製薬）280円
★4．リポビタンスーパー（大正製薬）230円
★5．ユンケル黄帝液（佐藤製薬）777円
★6．タケダC1000 Ca & Mg（武田食品）97円
★7．強力グロンサン（中外製薬）190円
★8．チオビタドリンク（大鵬薬品工業）130円
★9．リポビタンゴールド（大正製薬）300円
★10．ローズウォーター（エナジー・プール）600円

Computer Laws

Illustration:MIKIO SASAKI

コンピュータの法律

プログラムの画像に著作権は認められるか

◆

熊谷信太郎（弁護士）

人気ソフト『ドラゴンクエストIV』の発売元が，同ゲームの「攻略本」を出版しようとした出版社を相手どり，刊行物発行差止め仮処分を求めた事件*1がマスコミを賑わせましたが，この事件を題材に，コンピュータソフトの法的保護がどこまで及ぶかを考えてみたいと思います．

■著作物とは

その前に，日本におけるコンピュータソフトの法的保護の仕組みを整理してみましょう．コンピュータソフトは，著作権法により法的保護が与えられています．著作権の侵害があった場合には，使用の差止めや損害賠償請求をすることができます．そのためには，コンピュータソフトが「著作物」として認められなければなりません．

「著作物」は，著作権法によると*2，①思想または感情の表現であること，②創作的表現であること，③文芸，学術，美術または音楽の範囲に属するものとされています．どのようなものが「著作物」に該当するかということは，これも著作権法に例示されています．その例の中にコンピュータの「プログラム」が含まれており*3，「プログラムとは何か」という定義もされています*3．

■インターフェースは対象外

BIOS，OS，アプリケーションプログラムなどは，いずれも「プログラムの著作物」として法的に保護されると考えられています．ここで注意すべきは，ハードとBIOSの関係でのインターフェースは著作権法による保護を受けないと定められていることです*4．

インターフェースは，2つ以上の装置を連携して作動させるための，情報の伝達に関する取り決めですから，プログラムを作動させるために必要なものではありますが，あくまで，単なる約束ごとにすぎず，プログラムの本質的な部分ではないと考えて，このように規定しています．

■イラストも似ていればダメ

では，『ドラゴンクエストIV』の裁判では，ゲームの著作権についてどのような判断がなされたのでしょう．テレビゲームは，「プログラムの著作物」であることには異論がありません．この裁判での問題は，プログラムによりアウトプットされる画像を，プログラムそのものとは独立の「著作物」として保護できるかということです．

訴訟の対象となった『ドラゴンクエストIV』の「攻略本」では，ゲームの各場面の謎ときをイラストつきで解説しています．ここでは，プログラムそのものの著作権が侵害されたわけではなく，プログラムによってアウトプットされた画像が複製されて無断で使用されたことが問題になっています．

したがって，そのような画像をプログラムとは別個に保護すべきかということが主な争点になり，それにともな

ワンラインNEWS
NTTデータは東京霞が関ビルにショールーム「ファイズ」を開設．効果的なプレゼンテーションルームがあるほかNTTデータの最新技術を紹介する．

って，イラストが画像の無断複製にあたるかという事実認定の問題がでてきます．この仮処分事件では，裁判所はこれらをいずれも認めて，「攻略本」の出版の差し止めを命じました．

■画像は映画のようなもの

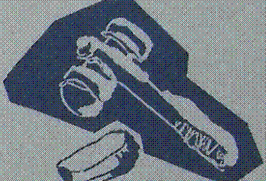
プログラムがアウトプットした画像について，裁判所は従来から「映画の著作物」として保護できるという判断を示しています[*5]．つまり，プログラムの画像が映画に似た視聴覚的効果をもつ著作物として認められていることになります．

テレビゲームの場合，ゲームをするものが独自に入力する部分があり，ゲームをやるたびに展開が異なるので，ストーリーが完成した映画とは少し違うという疑問もありますが，裁判所の考え方は，ゲームの場合もおよそのストーリーは決まっており，ゲームのルールに沿って入力する部分があっても「映画の著作物」として認めて，保護しようというものです．『ドラゴンクエストIV』の裁判では，仮処分事件であるため，なぜそのような判断をしたか理由は明らかにされませんでしたが，従来の判例に沿った判断をしたと思われます．

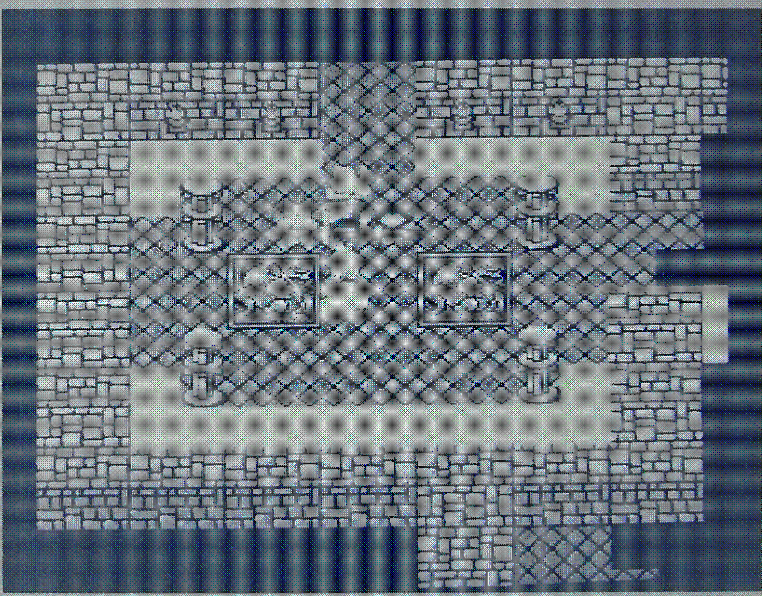
次に，「攻略本」に掲載されているイラストが『ドラゴンクエストIV』の画像の無断複製にあたるかということですが，これはゲームの画面とイラストに類似性があるかという，事実認定の問題になります．『ドラゴンクエストIV』裁判では，この点についても類似性を認め，著作権を侵害していると判断しています．

このように，日本の裁判所では，プログラムそのものとは別に，画像も著作権の対象となるという判例が確立しているといってよいでしょう．

＊1　東京地裁平成2年6月27日決定
＊2　著作権法第2条第1項第1号：「思想又は感情を創作的に表現したものであって，文芸，学術，美術又は音楽の範囲に属するものをいう」
＊3　著作権法第2条第1項10号の2：「電子計算機を機能させて1の結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したもの」
＊4　著作権法第10条第3項第二号では，「規約」という言葉でインターフェースに関して定めている．
＊5　たとえば，テレビゲームのパックマンの画像は，「映画の著作物」であり，プログラムそのものとは独立の「著作物」として保護できるとした判例がある．（東京地裁昭和59年9月28日判決）



ドラゴンクエストⅣの画面　（写真提供：株エニックス）



ないプログラマの情報収集にはもってこいの場所である．

2階はパソコン・ワープロ＆周辺機器フロアで，ハードディスクからプリンタ用紙まで，各社の周辺機器が取り揃えられている．

3階のビジネスソフトフロアには，内外各社のパソコン用ソフトが展示されているが，中でもIBM PC用のソフトにはアメリカのパソコンショップでもめったにお目にかかることのないMicrosoft Windows Software Development KitやCD-ROMのMicrosoft Programmer's Library，ユーティリティではSOURCER 486やTHE NORTON UTILITIES Ver.5.0など，プログラマならばちょっと気になるソフトが揃っている．

4階はビジネスシステムフロア，5階はCAD＆グラフィックスフロアで，それぞれ財務・会計システム，CAD・CGシステムなどが展示されている．また，5階にはNeXTも展示されている．

6階はイベントフロアとなっており，各種のセミナーやフェアが開催されている．

ワンラインNEWS
セガエンタープライズは「メガドライブ」を欧州市場へ輸出を開始した．欧州へは任天堂が「ファミコン」を輸出しているが米国や日本ほど普及していない．

Vertual Space Guide

Illustration: TOSHIYA SHIROTANI

バーチャルスペースガイド

田植えで欲情

C平　きのうレンタルビデオでデビッド・リンチの『ツイン・ピークス』借りて見ちゃった．

P子　ネクロフィリア（死体嗜好症）みたいなシーンがあって変態っぽいっていうヤツでしょ．ヤダッ．

C平　そうそう．でもああいうふうに徹底してカゲキな世界を築いていくってすごいよね．リンチは嫌いじゃない．『砂の惑星』もいい．『エレファントマン』はヤダけど．

P子　クラーい．あれも変態映画よネ．胎児みたいな宇宙人や空飛ぶ変態デブがでてきたりして．

C平　『砂の惑星』はぜんぜん期待してなかったんだ．原作がスペースオペラだし，それを中世ヨーロッパ風の世界に見立ててあるっていう話だったから．それで，T君の家にいったときに，「つまらないけど見る？」っていうから見ていたら意外と気にいっちゃってさ……．帰りにアキバによってＬＤ買っちゃった．

P子　T君っていえば，このあいだアキバでトランスかかえてヨタヨタ歩いてたわョ．でも，アノコ，映画とかの趣味はフツーだもんネ．変態はアンタよ．

C平　シルバーナ・マンガーノが出てるんだよ，『砂の惑星』には．

P子　ああ，ハゲのオバさんでしょ．アンタああいうの好きなの？

C平　ハゲのオバさんはないだろ．修道女かなんかの役だったんだよ．そのシルバーナ・マンガーノは若いころ『にがい米』っていうイタリア映画に出ていたんだ．

P子　なーに，それッ．

C平　うんと昔の映画だよ．モノクロだったもん．中学のときにテレビの深夜映画で流れてるの見てさ，イタリアの女のコが農繁期に田植えの手伝いをするときの出来事を描いてあるんだけど．

P子　えっ．日本にくる話？

C平　イタリアのタンボだよ．P子，このあいだイタメシ屋でリゾット食ってたじゃん．あっちにもちゃんと水張ったタンボがあるんだよ．

P子　そういえばイタリア料理って米つかうもんね．

C平　だろッ．その映画でシルバーナ・マンガーノがショートパンツはいて田植えをするの．そのフトモモがイロっぽくてさ，なんかオレ，コーフンしちゃってさ．

P子　ヤダ．バカっ．田植えで欲情してんの．でもって，マンガーノさんが忘れられないってーワケ．ナルホド．

C平　ちゅ，チューガクんときだヨ！

『チビまる子』返せよ！

P子　わたしは，あんまり好きじゃないわね．ああいう世界は．SFっぽいので好きなのはル・グィンくらい．

C平　ル・グィンは，SFっていうのがおかしいくらいで，岩波の少年少女向けのシリーズになっているよね．

P子　『ゲド戦記』の３部作ね．あれもよかった．でも，わたしがもっと好きなのが『闇の左手』．両性の人たちが住む寒い惑星の話．

C平　P子もけっこうクライじゃない．

P子　ゲッ．たしかに雪と氷に閉ざさ



P子　某ソフトハウスで企画営業を担当している．26歳．
C平　某メーカーのシステムソフトウェア開発部門に所属．25歳．

れた惑星の話になっているから印象として暗くはあるわね．でも，山岸涼子のマンガみたいな魅力もあるし，人の心のシミュレーションがすごくよくできている．

C平　そうだね．SFってサイエンスじゃなくて，シミュレートフィクションにしたらいいんじゃない．もう科学小説っていう時代じゃないもの．小説のなかで，時間・空間的にシフトの大きいものをSFと呼んだらいいんだよ．

P子　ル・グィンは，根底的なところで性を意識しているの．両性具有の話もそうしたテーマのひとつなの．

C平　オレ，その本は読んでないから．

P子　じゃあ，シフグレソルなんていってもわからないんだ．

C平　なんだよそれ．

P子　人と人の精神的緊張感を形成するある概念っていうのかな．恥の意識とか，プライドとかに類するものだと思うんだけど，そういう概念を小説の中で作りあげているわけ．とにかく読んでみなさいよ．

C平　そのうちに貸して．あッ，『チビまる子』返せよ！

P子　なによ，突然．もう読んだんでしょ．いつでもいいじゃない．

C平　だーって，テレビの『チビまる子』つまんないんだもん．タラコはでしゃばりだし，だいたい，貸してあるのは『リボン』をバラしてファイルした貴重品なんだから．

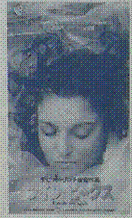
P子　いい歳してなにいってんのよ．まだ『リボン』なんか読んでるわけ．ビョーキよ．

DATA

■ツイン・ピークス(1990年・米)

監督：デビット・リンチ
出演：カイル・マクラクランほか

ツイン・ピークという小さくて平和な町に少女の水死体が流れ着く．そこからはじまる平穏な生活の崩壊……．米国のテレビシリーズをまとめて，日本ではビデオでリリースされた．『ブルーベルベット』にも見られたように，平穏な日常をひきはがすような手法でリンチ監督の世界が展開される．

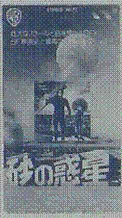


ワーナー・ホーム・ビデオ

■砂の惑星(1984年・米)

監督：デビット・リンチ
出演：カイル・マクラクラン

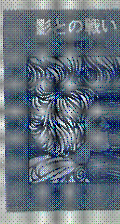
フランク・ハーバートの長編SF「デューン・砂の惑星」の映画化作品．超空間移動を可能にする香料メランジを産する惑星アラキスを舞台に，教会，王家，原住民などが争いを繰り広げる．「空飛ぶ変態デブ」「胎児のような宇宙航行士」「ハゲの教母」「巨大な虫」など出演者（?）は多様であり，それらが奇怪な世界を築いている．主演はリンチ作品常連のカイル・マクラクラン，『ブレードランナー』でレイチェルを演じたショーン・ヤングも出演している．



ワーナー・ホーム・ビデオ

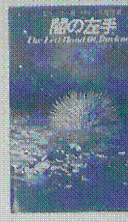
■ゲド戦記(岩波書店)

影と戦う魔法使いゲドを描いたアーシュラ・K・ル・グウィンの3部作（第1部『影との戦い』，第2部『こわれた腕輪』，第3部『さいはての島へ』）．アースシーという世界を舞台に展開される冒険は，魔法の戦いあり，ダンジョンありで，さながらアドベンチャーゲームだが，光と影，生と死などの根元的なテーマをふまえた空想叙事詩になっている．



■闇の左手(ハヤカワSF文庫)

1969年に発表されアーシュラ・K・ル・グウィンのSF界における名声を不動にした作品．宇宙の辺境に閉ざされた両性をもつ住民が住む惑星<冬>を訪れた宇宙連合の使者アイの旅行記の形をとる物語．異国人と対面するアイの不安と孤独感を，<冬>住民の心理や風俗の抑制のきいた描写によって浮かび上がらせる．ル・グウィンの真骨頂ともいうべき一編だ．



創刊記念プレゼント

毎月HANDY98を1名様にプレゼント！

2 Microsoft C 6.0 コンパイラ



1名

マイクロソフト(株)提供

3 Turbo C++ Profetional



1名

ボーランド・ジャパン(株)提供

創刊号から創刊第３号で実施！
（商品は月により多少異なります．）

7 通信ソフト まいとーく



1名

㈱インターソフト提供

1 HANDY98 (PC-98HA)



1名

11 NeXT テレホンカード



20名

キヤノン(株)提供

●応募は，本誌に添付のアンケートはがきをご利用ください．（1991年１月10日到着分有効）　●応募者数多数の場合は抽選になります．なお，

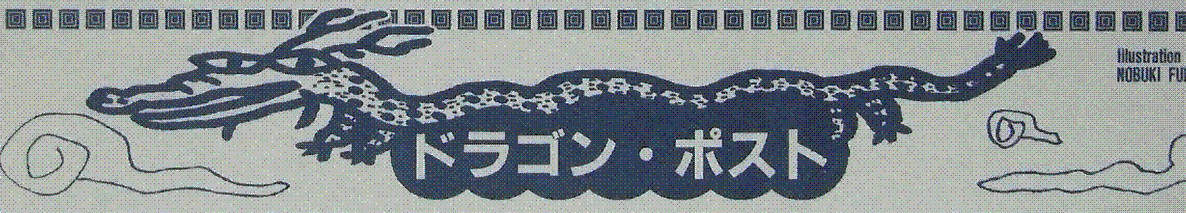


Illustration NOBUKI FUKU

ドラゴン・ポスト

この欄は，読者からのお便りを中心に構成します．本誌に対するご意見・ご感想などをお待ちしております．今回は，本誌の企画にあたって実施したアンケート調査から，「こんな雑誌が欲しい」というご意見を紹介します．

■こんな雑誌が欲しい

●プログラミングの「へそ」のようなものの指摘．（横浜市・浅井克彦）

●当り前のことは専門書が幾らでもある。トリッキィなことやハードとのからみでの特殊な体験など，読者との対話が主体となる記事が欲しい．（埼玉県・有末進一）

●右も左もCという状況ですので，私のようにCをかじってはみたけれど，TURBO-PASCALがやっぱり好きだという人間の読む雑誌がないのがさみしい．（東京都・石田圭祐）

●単なるプログラム誌ではなく，プログラマの悩みや苦しいこと，楽しいことなど，これからの参考にできるようなことを多く扱って欲しい．（東京都・杉山陽一）

●冗長すぎない文章．自分の専門分野のより深い情報，専門外の分野の入門，概要情報，よそのプログラマの待遇など．（愛知県・尾島俊之）

●私の場合もそうですが一つの言語に精通しかけると他の言語に目を向けなくなります．いろいろな考え方（アルゴリズム）を身に付けるためにも広く深く編集して欲しい．（栃木県・鈴木幸則）

●専門分野をきちんと抑えること，時節にあったテーマを追求すること．読者が気楽に参加できる欄があること．特に楽しめる所が一つくらいないと続けて読む気になれない．（埼玉県・柳橋岳夫）

●専門馬鹿のための雑誌ではなく，これからの社会を背負



当選者の発表は発送をもって代えさせていただきます.

っていく立派な社会人のプログラマー養成のための雑誌.(東京都・山原雅史)

●ノウハウの公開の主導となっていただきたい. 読者の意見交換が活発に行われるような場を作り出して欲しい. 中級以上のアルゴリズムのコンクールなどいいのでは? (東京都・吉住英二)

■初心者向けの記事を望む

●BASICでお遊びする程度の私にとっては, きっと高度な雑誌なんだろうと思うけど, 私にもこれからはそういうものにも取り組んでいこうと思わせるような初心者向きページも欲しい. (京都府・上野定子)

●全てを網羅するのは無理でも, いきおい「専門的=難解」なるところをわかりやすい記事もあるといいですね. (千葉県・小圷章)

●毎号毎号初めてその雑誌を買った初心者を意識して紙面作りをしてもらいたい. (埼玉県・鳥羽幸信)

■初心者向けの記事はいらない

●最近のパソコン誌は初心者向けばかりで読むとこがないので期待している. (東京都・藤島聡)

●初心者をよせつけないきびしさを持った記事. ただし, やみくもに拒絶するのではなく勉強する意志のある人には門戸を開くような…そんな記事. むずかしいでしょうね. (神奈川県・大島誠)

●プロのための手ごたえのある記事. (神奈川県・季高俊秋)

●入門書を読めば済むような記事を連載したりしないこと. (茨城県・堀野眞一)

■機種・言語に限定されない記事を

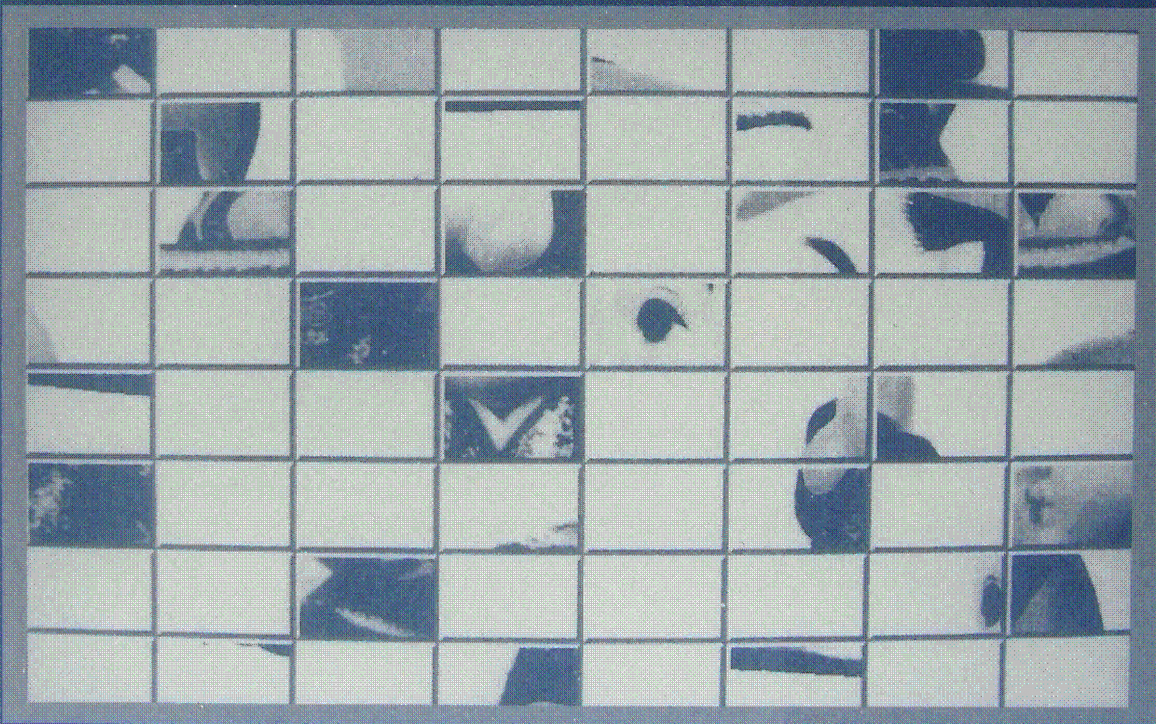
●一般的な入門記事, 解説に加えて, プログラマにとっては常識だが, 一般の雑誌には出てこないようなテクニックを紹介する記事が欲しい. また, 98, IBM, MACといった主要機間のプログラム, データの互換性, 移植等に関する基本的な解説記事もあると便利. (青森県・高岸聖彦)

●PC-9801にかたよらないで欲しい. これからのOS「OS-9000」関連を記事にして欲しい. (東京都・安達洋治)

●特定の機種, 言語処理系に限定されたものでない, 汎用的な記事を望みます. (大阪府・清水好三)

クイズ・ランダムスクリーン

ある１枚の画像を64分割し，ランダムに並べ替えてあります．
さて，もとの画像は何を写したものでしょうか．
ヒント：お蕎麦屋さんにあることも……．



正解者の中から30名様に本誌表紙イラスト（日暮修一画）入り
月刊プログラマーズ・ページ特製オリジナルマグカップをさしあげます．
応募方法：本誌に添付のアンケートはがきをご利用ください．
締め切り：1991年1月10日（到着分有効）
なお，当選者の発表は発送をもって代えさせていただきます．

月刊プログラマーズ・ページ
プログラムサービスのお知らせ

月刊プログラマーズ・ページでは，本誌に掲載したプログラムのサービスを行っています．
入手方法には，次の２つの方法があります．

1・BBSでのダウンロード

1. 03-5276-1297に電話をかけます
2. 接続後IDの入力を求めてきますので，GUESTと入力し，リターンキーを押してください．
3. お知らせを表示後，メインメニューのコマンドプロンプトが表示されます．
4. Tと入力するとファイル転送システムに移行します．
5. ダウンロードする目的のプログラムがあるボードの番号を入力します（＊と入力するとボードの一覧が表示されます）．
6. Bと入力するとファイルの一覧が表示されます．
7. ダウンロードするファイルの番号を入力します．
8. ご使用の通信ソフトにあわせて転送プロトコルを番号で選択します（?でプロトコルの一覧が表示されます）．
9. ファイルの転送が終了したら，0と入力し回線を切断します．

2・直接購入

希望メディアサイズ（5インチまたは3.5インチ），プログラムサービス希望の本誌号数をご記入のうえ，現金書留か郵便振替（東京4-154493），または料金分の切手を同封のうえ，お申し込みください．

●送り先：〒102　東京都千代田区平河町2-4-14
平河町KSビル
株式会社 翔泳社
月刊プログラマーズ・ページ　プログラムサービス係
●価　格：1,500円（消費税，送料，ディスク代込）

・本サービスは入力の手間を省くため，本誌に掲載したソースファイル，実行ファイルを提供するものです．基本的には，掲載時のままでサービスいたしますので，プログラムのなかには，ライブラリファイルのように単独では実行できないものもあります．

・掲載後に発見されたバグなどについては，本誌でお知らせします．また，マニュアルなどは付属していませんので，該当記事をご参照ください．

・プログラムの動作環境，対応機種などについては該当記事でご確認ください．

Programmer's Page
MONTHLY MAGAZINE FOR PROGRAMMERS

月刊プログラマーズ・ページ2月号
（1月18日発売）

特集

徹底解説 Windowsプログラミング

昨年米国を襲ったWindows 3.0の波が日本にやってくる．MS-DOSソフトウェアとは異なるWindowsプログラミングの需要が確実に増えようとしている．新しいWindowsプログラミングの世界を，概念，環境，技法などから多面的に，そして，ビジュアルをふんだんにつかって徹底的に解説します．

月刊プログラマーズ・ページ　1991年1月号
1991年1月1日発行

発行所　株式会社翔泳社
東京都千代田区平河町2-4-14平河町KSビル4F・5F
TEL 03-263-0447　〒102
発行人　長廻健太郎
編集長　清水　隆
副編集長　鈴木嘉平
編集スタッフ　杉野貴美子＋上田純美礼＋芳山充弘＋勝山まゆみ
デザインスタッフ　佐々木幹夫＋SE（翔泳社デザイン研究所）
広告スタッフ　相澤正夫＋澤本良知＋石崎健治
協力　瀬川清＋根本裕三郎
写植　株式会社東巧社
印刷　凸版印刷株式会社
定価　980円（本体951円，税29円）

●「プログラマ」と銘打ったパソコン雑誌は，本誌がはじめてであろう．パソコンがパーソナルな機械であることによって，いちばん恩恵を受けているのはプログラマではないだろうか．ターゲットをどこにおいても，プログラミングは自分の気に入っているパーソナルな環境で行えるようになった．お仕着せの機械や環境で居心地の悪い思いをしながらプログラムを組まなくてもよくなったのだ．成熟してゆくパソコンプログラマの世界を映し出してみたい．（TA）

●自宅で使用しているマシンをPC-9801VX2からIBM PS/55zに変更した．WIN3やMS-C6.0など，アメリカで話題になっているソフトをなんのタイムラグもなく使用できる快感はこたえられない．数え切れないほどあるON-LINEソフトを使うのも心楽しく，いそいそとCompuServeにアクセスする毎日，FAXオーダーのやり方も覚え，PC環境は日々整いつつある．しかし，お金が……〈^_^;〉　ボーナスを心待ちする今日この頃であった．（嘉平）

●ノートパソコンブームに思う。持って歩くことを前提にしたらもっといろんなソフトやハードがあっていいはず。競馬の馬券購入支援、将棋の名対局再現、麻雀勝ち負け記録、今週のテレビ番組チェック。ベネトンカラーのノートパソコンに詰め込んでお出かけなんていかがでしょ？（kos）

●「こんな雑誌なんだけどどう思う？」と質問してみた．S君「おもしろそうだけど……やっぱマニアぽいんじゃない？」，Y子「よくわかんないけど，それって今流行のおたくの雑誌？」と，反応はさまざま．『上をねらったばっかりに→パソコンに手を染め→プログラマになってしまう』人が増えるかもしれないし，プログラマも不足してるっていうし，ちょっとへんな雑誌があってもいいんじゃないかと思うのは私だけでしょうか．（ま）

●その昔私が居住していた汎用機の世界では，「プログラマ」という言葉は「SE」の下に位置づけられ，2つの言葉には明確な違いが存在した．事実，「プログラマですか」と聞かれた時，「いいえ，SEです」と必ず訂正するような微妙なニュアンスの違いがあった．だがパソコンの世界では，この2つを明確に区別するのが難しく，また逆に，プログラマという言葉のもつ職人的な響きを好んで使う人が結構多いというのが現状のようだ．（K）

●「今，駐車場に空きありますか？」「ないけど，半年位待てば空くかも知れないから予約を入れる？」「いくらですか？」「月3万5000円，この辺の相場だよ」「……えっ！！」と，たいへん天気のいい某日曜日午後の不動産屋さんでの会話．こんなことなら多摩地区に住んでるほうがよかったと，BMWのM7を手に入れた友人を恨めしく思い，引越しを後悔する今日この頃です．ああ，新車が遠のく……（よ）

●編集後記というものは，全ての作業が終了しビール片手に書くものだと考えていた私が甘かったというべきでしょうか．この殺人的スケジュールをクリアし，美酒に酔いしれるのは何時のことやら....エディトリアルとは体力です．最後に短いスケジュールの中で素晴らしいイラストレーションを書き上げていただいた日暮修一氏と，表紙モデルの桜井久直氏，そして数多くのアドバイスをしてくれたデザインスタッフに感謝いたします．（MIKIO）

愛読者カードご記入のお願い

本誌をお買い上げいただき、誠にありがとうございます。
お手数ですが、今後の誌面づくりの参考にさせていただきたいと存じますので、
各欄にご記入の上、ご返送くださるようお願い申しあげます。

月刊プログラマーズ・ページ READER'S SURVERY CARD 愛読者カード

本誌の号数　　年　　月号

本誌を知った理由
□1.書店店頭　□2.他社出版物（　　　）
□3.小社出版物　□4. DM　□5.口コミ

パソコンの所有形態（機種名）
□1.個人所有　（　　　）
□2.職場にある（　　　）
□3.持っていない

よく使用するソフトウェア

よく使用するOS

今月号でおもしろかった記事（いくつでも）

今後取り上げてもらいたい特集のテーマ

貴方が選ぶベスト3

パソコン雑誌	①	②	③
ソフトウェア	①	②	③
パソコン本体	①	②	③

本誌についてのご感想、お気づきの点、その他どのようなことでも結構ですので、ご意見をお聞かせください（「ドラゴン・ポスト＝読者の声」欄に掲載させていただく場合もあります。ペンネームをご希望の方はあわせてご記入ください）。

月刊プログラマーズ・ページ 広告資料請求カード

8	会社（学校）名		9	所属（学部）		10	従業員数	

※下記の該当する番号をご記入ください。

11	業種		12	職種		13	パソコン利用状況	

業種
①製造業　②金融・保険・不動産業　③商社・卸・小売業
④建設業　⑤通信・運輸　⑥システム・ソフト
⑦その他サービス業　⑧公務員　⑨その他

職種
①コンピュータ（ハード）技術者　②コンピュータ（ソフト）技術者　③エレクトロニクス技術者
④プログラマ　⑤その他の技術者　⑥専門職
⑦事務・営業職　⑧経営管理職　⑨商工サービス
⑩教職　⑪学生　⑫その他

パソコン利用状況　①会社で利用　②個人で利用　③会社でも個人でも利用　④利用していない

●希望される広告資料　　年　　月号

希望項目 ＼ 広告資料請求No.												
A 購入希望	A	A	A	A	A	A	A	A	A	A	A	A
B 説明希望	B	B	B	B	B	B	B	B	B	B	B	B
C 資料希望	C	C	C	C	C	C	C	C	C	C	C	C

定期購読のご案内

綴じ込みの振替用紙を使って、本誌の定期購読をお申し込みになれます。
1年間12冊＝11,760円（本体11,417円/税343円）
郵送料サービス

創刊記念 HANDY 98 プレゼント

郵便はがき

料金受取人払

麴町局承認

3908

差出有効期間
平成４年11月
30日まで

切手不要

102-00

東京都千代田区平河町2-4-14
平河町KSビル

株式会社 翔泳社
マーケティング部 行

月刊プログラマーズ・ページ READER'S SURVEY CARD 愛読者カード

氏名	性別 □1男 □2女	年齢 歳
住所 〒		電話 （　　） －

職業
□1ソフト技術者 □2ハード技術者 □3技術系会社員
□4事務系会社員 □5営業系会社員 □6教師
□7一般公務員 □8経営管理職 □9理系学生
□10文系学生 □11その他（　　　）

ユーザー区分
□1 プロ
□2 マニア
□3 一般ユーザー

創刊記念プレゼント記入欄	クイズ・ランダムスクリーン回答欄
希望商品番号	答

郵便はがき

料金受取人払

麴町局承認

3910

差出有効期間
平成４年11月
30日まで

切手不要

102-00

東京都千代田区平河町2-4-14
平河町KSビル

株式会社 翔泳社
月刊プログラマーズ・ページ
広告資料部 行

＊当社で記入

1	SE	2	受付No.	

月刊プログラマーズ・ページ広告資料請求カード

3	会員番号	

＊すでに会員番号をお持ちの方はご記入ください。

4	フリガナ 氏名	姓　　名	5	年齢	歳

6	自宅住所	〒　　電話（　　）
7	会社住所	〒　　電話（　　）

資料送付希望先の番号に〇印をつけてください。

R!

の

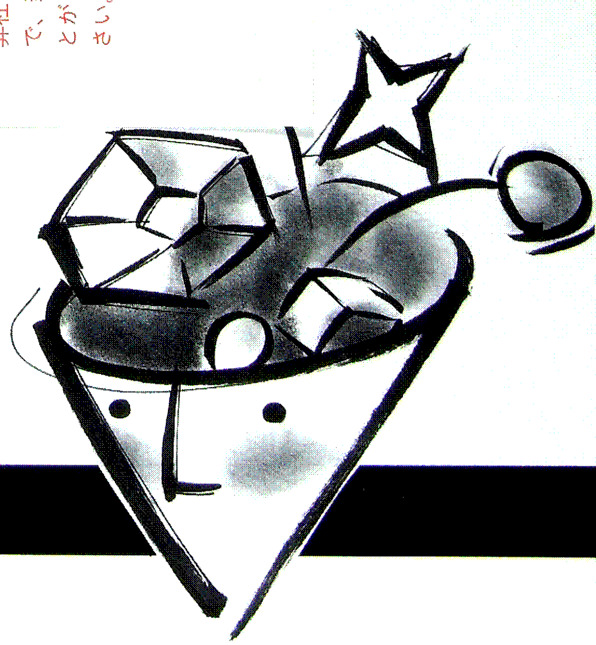
合

なのが、気力と体力に自信がある

かくスッゴイ大変なんです’とは、
ームソフトプログラマーの第一声。
帰れれば恩の字で、泊り込みなん
っちゅう、というのは決してオー
はなさそうだ。チームを組んでの
から、一人の遅れはみんなの迷惑。
たしていると同業他社に先を越さ
行を逃すという悲惨な結末が待っ
だけだ。やり遂げるまで、ただひ
突っ走るのである。自由でルーズ
が持てて、世間とは逆さまの生活
ても元気で、‘絶対完成させてや
という根性のある人でないと、こ
は難しい（やり始めたら止まらな
、という理由もあるらしいが）。

ば生き残れない。「遊び心」には際限がない。‘もっと面白いゲーム’‘もっともっと痛快な遊び’という買い手の要求の何歩も先を走れなければ、ユーザーを引っ張り続けることはできないからだ。

すべては「想像」から始まる。‘こんなゲームがあったらいいな’‘こりゃ画期的だ’と、アイディアをほとばしらせる。欲しいものは宇宙的に果てしないセンスとユーザーよりウワテの遊び心だから、そこには学歴も技術もキャリアも無用だと言える。事実、20歳前後の‘ヤワラカ頭世代’が大活躍している企業も多い。少年少女とおんなじ夢をいつまでも見られる、そんな透明さもこの仕事の宝だ。



通常払込料金加入者負担

払込通知票

各票の※印欄は、払込人において記載してください。

口座番号	東京 4 ▒ 154493
金額	11760
加入者名	株式会社　翔泳社
料金	払込み／特殊　円
払込人住所氏名	※（郵便番号　　　） TEL（　　　　）
備考	
受付局日附印	

記載事項を訂正した場合は、その箇所に訂正印を押してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵政省）

切り取らないで郵便局にお出しください。

通常払込料金加入者負担

払込票

口座番号	東京 4 ▒ 154493
加入者名	株式会社　翔泳社
金額	11760
払込人住所氏名	※
備考	
受付局日附印	

郵便はがき

102-00

東京都千代田区平河町2-4-14
平河町KSビル

株式会社 翔泳社
月刊プログラマーズ・ページ
広告資料部 行

2	受付No.	

ーズ・ページ広告資料請求カード

＊すでに会員番号をお持ちの方はご記入ください。

名		5	年齢	歳

6	自宅住所	〒　電話（　　）	資料送付希望先の番号に○印をつけてください。
7	会社住所	〒　電話（　　）	

綴じ込みの

1

創刊記念 HANDY 98 プレゼント

□7一般公務員　□8経営管理職　□9理系学生
□10文系学生　□11その他（　　　）

□3一般ユーザー

創刊記念プレゼント記入欄	クイズ・ランダムスクリーン回答欄
希望商品番号	答

WE WANT POWER!

オブジェクト指向の就職読本 第1回

▶会社が求めるスタッフ◀
ゲームソフト開発メーカーの場合

「仕事が遊びか、遊びが仕事か。楽しそ！」という羨望の目で見られ、趣味と実益を兼ねられるのではと安易に飛びつかれてしまいがちな、ゲームソフト開発メーカー。しかしその実態は、想像以上に大変でやりがいもある世界なのだ。今回の特集では、ゲーム開発の現場の話をまじえながら、会社がどのような才能や人物像を求めているのかを探ってみることにしよう。

●まずは、頭がアイディアの宝庫である人だ

現在、普及台数2,200万と言われるファミリーコンピュータ。ソフト開発メーカーは、ニーズを先読みして、ユーザーの期待以上の大傑作を創っていかなければ生き残れない。「遊び心」には際限がない。‘もっと面白いゲーム’‘もっともっと痛快な遊び’という買い手の要求の何歩も先を走れなければ、ユーザーを引っ張り続けることはできないからだ。

すべては「想像」から始まる。‘こんなゲームがあったらいいな’‘こりゃ画期的だ’と、アイディアをほとばしらせる。欲しいものは宇宙的に果てしないセンスとユーザーよりウワテの遊び心だから、そこには学歴も技術もキリャアも無用だと言える。事実、20歳前後の‘ヤワラカ頭世代’が大活躍している企業も多い。少年少女とおんなじ夢をいつまでも見られる、そんな透明さもこの仕事の宝だ。

●夢をたかちにできるだろうか

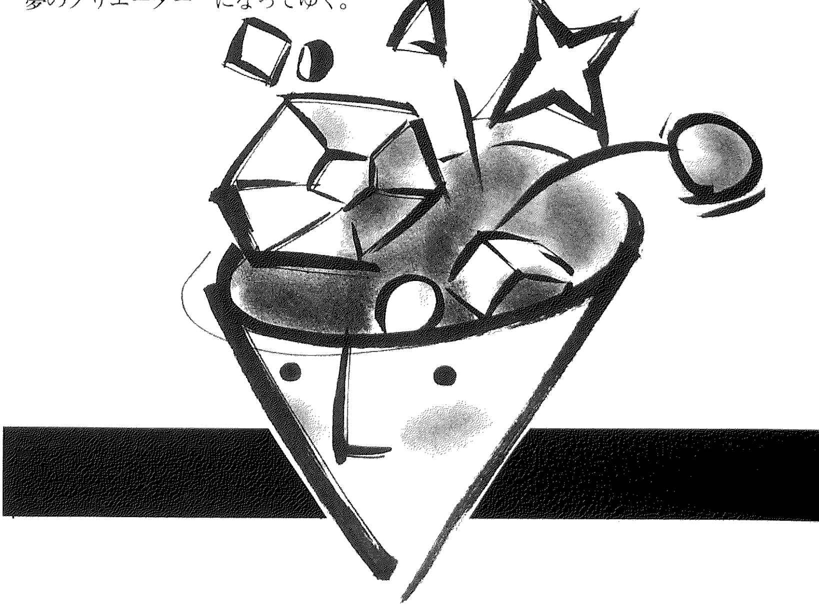
ハッキリ言って、夢を見ることにかけては子供の方が優っているかもしれない。大人として、プロとしての本領発揮はここから。つまり「想像」から「創造」するテクニックなのである。

通常7～8人のグループで、あーだこーだとブレーンストーミングを重ねながら、夢の実現に向けて試行錯誤を繰り返す。先端のテクノロジーを駆使する。先端で物足りなければ‘最先端’を生み出す。時代との競争心、技術のあくなき追究心、強力なチームワークと開拓者魂。それらをうまく使いこなして初めて「完成」という最大の喜びが得られる。

立派に商品となったゲームソフト。夢を手にとって、夢で遊んで。そうした醍醐味を味わうごとに、ひとまわり大きな‘夢のクリエーター’になってゆく。

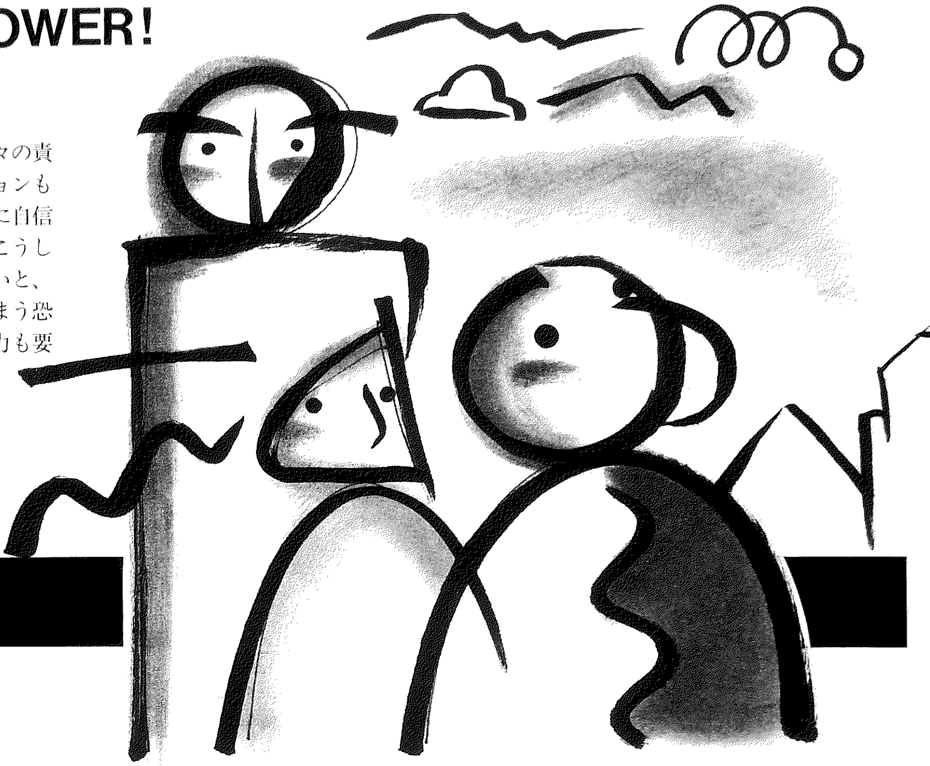
●重要なのが、気力と体力に自信があること

‘とにかくスッゴイ大変なんです’とは、あるゲームソフトプログラマーの第一声。夜中に帰れれば恩の字で、泊り込みなんてしょっちゅう、というのは決してオーバーではなさそうだ。チームを組んでの作業だから、一人の遅れはみんなの迷惑。もたもたしていると同業他社に先を越され、流行を逃すという悲惨な結末が待っているだけだ。やり遂げるまで、ただひたすら突っ走るのである。自由でルーズな時間が持てて、世間とは逆さまの生活を続けても元気で、‘絶対完成させてやる！’という根性のある人でないと、この仕事は難しい（やり始めたら止まらなくなる、という理由もあるらしいが）。



WE WANT POWER!

●気概と誇りもなくてはならない
　分業の同時進行作業のため、各々の責任は重く、相互のコミュニケーションも必要となってくる。自分のパートに自信と誇りを持って‘こうしたい’‘こうしてほしい’と堂々と主張ができないと、慣れ合いと妥協の作品になってしまう恐れがある。協調性と共に、折衝能力も要求される。



君のスターティンググリッド

GET THE POLE POSITION

　ゲームソフト開発を手がけるには、どの程度の技術レベルが必要なのだろうか。編集部では、ゲーム開発者への取材をまじえながら、次のような感触を得た。それは､「アセンブラ、あるいはC言語を一通り経験していれば、それだけで十分」といったレベルである。つまり、文字や言葉で言い表せる技術力をいかに十二分に発揮できるか、その技術をどう応用できるかといった、形にならない才能こそが必要なのである。あれこれ技術的条件がない方が、逆にシビアだと言えるだろう。

　さらに‘欲しい人材’について、声を集めてみた。
「真面目な技術者よりも、‘いい加減で変なヤツ’がいい｣(ユニークなアイディアと柔軟な仕事ぶりが期待できる)
「ハードスケジュールに対応できる、強靭な肉体と精神力を持った人が要るね」
「湧き出た発想を、確実にプログラミングできる技術と根性と意地のある人でないと」
「‘人と同じことをやるのは死んでもイヤダ！’という、独創性の持主」
「ゼロから何かを創り上げていく喜びにとりつかれてしまった人」
「自分が手掛けたゲームで、ユーザーが一喜一憂する‘やり甲斐’に酔える人」
「遊び心とプロ意識、ルーズさと責任感を同居させられる人」

　さて、あなたはいかがだろうか。ともあれ、ゲームソフトの開発を志すなら、ふだんからアイディアを生む感性と体力だけは、最低限養っておかねばなるまい。

事情通の眼

現場の声

さいきんのソフトハウスの雰囲気について

ソフトウェア開発の現場は、どのように変わりつつあるのだろうか。年々増え続けている‘プログラマーのための職場’。各社が競って優秀な人材を求めていることで、逆に求職者には入社後かなりの好条件、好環境が準備されている。

週休2日制は今や常識。フレックスタイム制も当り前になりつつある。住宅手当、独身寮完備で、休暇はこぞって120日以上、残業時間もどんどん短縮されつつある。気になる研修制度についても、未経験者が立派な技術者になるまでとことん面倒を見てくれる。

今回は、最近転職した2人に、ご自慢の社風や耳より情報について語ってもらった。

K氏：僕は大手の通信機器メーカーに5年いたんだけど、辞めようと思ってもすぐには辞めさせてくれなかった。アトガマが決まらないとね。でもありがたいことに、今の会社は「人材登録システム」があって、入社希望時期にかなりの幅を持たしてくれたんだ。余裕を持って準備ができたよ。

Y氏：僕のとこは入ってからの対応が幅広かった。っていうのは、求人欄見てると「＊＊が出来る方」っていう条件付きが多いのに反して、うちはレベルに合ったワークフィールドがあるんだ。経験者も新米も時期を同じくして「即戦力」になれるってわけ。そこからも頑張り具合でトントンとステップアップできる。チャレンジ精神を評価してくれるところが嬉しいね。

K氏：「嬉しい」って言えば、入社早々1人1台ラップトップがもらえたんだ！

Y氏：うん。そういう会社、幾つか出て来てるみたい。うちの‘全員支給’は「弁当」。独身者には結構有難いもんなんだ、コレ。

K氏：そりゃいいね。僕んとこは社員がつくる「食堂委員会」ってのがあってサ、社員食堂のメニューを検討するんだ。味や栄養、もちろん価格も。

Y氏：社員が運営する組織って結構充実してきてるようだね。友達が行ってるソフトウエア開発の会社は、毎月給料日が「社内運営日」っていうのに充てられて、全員で業務の改善とか勉強会とか社内レクの話なんかして1日終えるんだって。

K氏：全員会議か。この業界は‘タテ割り’があんまりないよね。上下関係なくみんなで造っていこうっていう意識を持ってる。「フツウ」の会社には収まりきれなかったいい意味での野心家が多いんだろうね。

Y氏：そう。‘ああしたい、こうしたい’‘こんな風になればいいのに’という気持ちを一杯持った人が造っていくから、自ずと社員のニーズにぴったり合うように変わっていけるんだ。

K氏：転職者が一番不安な技術面でも、何カ月もマンツーマンで研修してくれるとことか研修センターがドーンとある会社も増えてきてるもんなア。

Y氏：うちは、仕事と余暇の両方を一か所で満足させてしまおうという贅沢な「インテリジェントビル」を、今建ててる。設備投資もすごいよね、最近。

K氏：人に投資する体制だってある。「独立万歳!!」で、将来経営者になりたい人のバックアップまでしてくれるんだ。この会社の懐の深さに、ヤル気が倍増したよ。

Y氏：伸びる一方の業種で、どんどん働き易くなっていく。しかも中途採用の道が年々開かれて、自分の夢が確実に近づく。前向きな転職だったら、みんなに勧めたいくらいだよ。

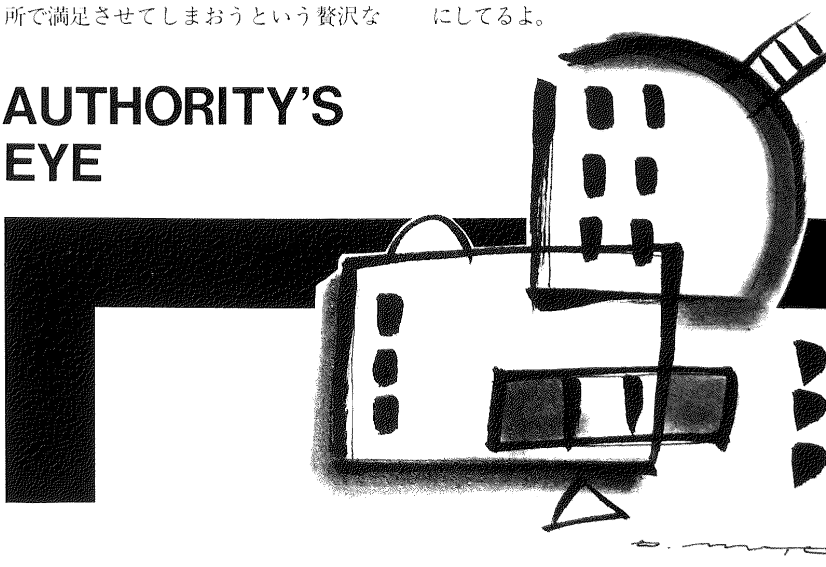
K氏：その通り！　でもすぐ目移りしちゃったら困るけど。

Y氏：そんな奴は元から前向きじゃないよ。それに改善案を反映してくれる社風にどこもなりつつあるんだから、会社に自分を合わせるんじゃなくて、自分に合った会社にしていくことだって十分可能なんだ。今度入ったとこってそうだろ？

K氏：そうだな。よ〜し！　お互いにバリバリ頑張って、も少したったらサア、また会って、自分や会社の成長ぶりをせいぜい自慢し合おうや。

Y氏：あははッ。そりゃいいや。楽しみにしてるよ。

AUTHORITY'S EYE



［ここだけの話］
実は転職を考えてるんです

だれもが何度かは考える転職。だが、その理由や動機は正当なものだろうか。また、転職にあたっての状況や事態は適切だろうか。転職者の疑問や不安に答えるQ&A。

Q：とにかく行き詰まっているんです。仕事は毎日同じようなことの繰り返し。上司も僕にあまり期待してないみたいだし、職場に活気がないんです。イヤイヤ勤めるより、心機一転新しい会社に移ろうかと考えているんですが。(27歳。大手電気メーカー勤務)

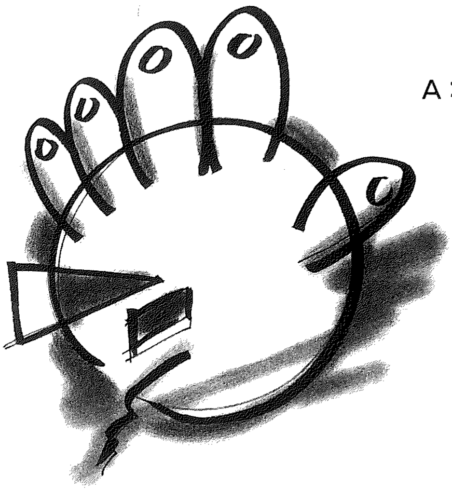
A：転職に多いに結構!!　でもハッキリ言います。今のあなたでは「不採用」でしょう。技術者の中途採用は年々増え、中でも電子、電気分野の技術者は引く手あまたの傾向にあります。しかし、技術があればいいかというとトンデモナイ。企業が欲しいのは‘逃げ腰の転職者’ではなく、‘意欲に燃えた自己開拓者’なのです。

1．自分の技術に自信を持ち、転職してからの目的意識がはっきりしている。
2．専門能力を応用する柔軟性とパワーがある。
3．独創性、新分野へのチャレンジ精神が旺盛である。

これが、企業の人事担当者が異口同音に述べる「採用の最低条件」です。

行き詰まりを感じるのは、向上心がある証拠。では自分は何をやりたいのか、転職先でどんな気概でやっていくのかを、あなたの中でしっかり確立した上で、胸を張って新しいドアを叩いてください。





Q：僕はこの３年間というもの、ひたすら自分を「スペシャリスト」に育ててきました。まわりから‘オタク’と言われようとも、黙々とSEとしての専門技術を研いてきたんです。この力をもって、違う企業で「エキスパート」としてやっていきたいのですが。(23歳。ソフトウェアハウス勤務)

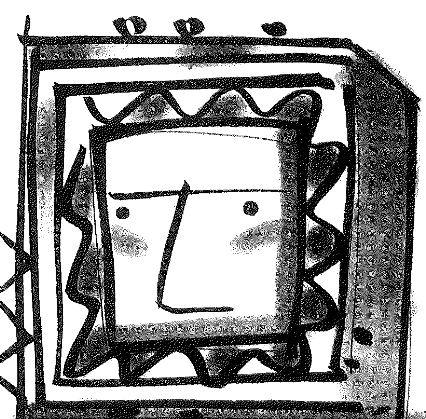
A：３年間もオタクして頑張ったわけですから、その専門知識たるや相当のものでしょう。そうした自信は貴重ですし、即戦力として注目してくれるはずです。ただし今、専門技術を武器とするいわゆる「職人」を求める時代は終りました。上級SEになるには、その技術をもとに他者あるいは協力会社とのコミュニケーションをとりながら、提案、折衝交渉できる能力が必要となってきているのです。

あなたの「対人関係能力」はいかがですか？　自分の技術や意見をうまく伝え、和を保つことが出来ますか？‘オタク族’から‘プロジェクトリーダー’へと、自分の意識を改革して行くことをお勧めします。

Q：大手メーカーがいますぐ欲しいと考えている職種を教えてください。(29歳。いますぐ転職し太郎)

A：１位：電子、電気回路設計
２位：電子機器、コンピュータ
３位：ＦＡメカトロニクス
４位：生産管理、品質管理、制御システム
５位：機械開発、機械設計
６位：機構開発、機構設計
７位：情報、通信ネットワークシステム
８位：画像システム、CAD／CAM／CAE
９位：その他研究開発、設計、SE

というデータがでています。共通して言えることは、「即戦力＋企業の体質強化改善」即ち、すぐ使える人に加えて、少々経験が浅くても意欲に燃えた人材は確保し、将来に向けて育てていこうということです。プログラマーのニーズは、今後ますます多種多様化するのは必至。転職を念頭に技術を研くもよし、転職後の研修などで目的にあった修行を積むもよし。要はあなたの目的とやる気です。




新たなる未知の果実を生み出すために
先端的頭脳の種子を求めています
人材募集
新卒者・経験者募集
たとえば、人間が初めて絵を描いたとき、小さな平面に表現される世界にどれほどの感動を覚えたことでしょう。初めて道具を使ったとき、自分の身体以外の可能性にどれほど驚いたことでしょう。未知のものに形を与える。私たちはその喜びを胸に歩んで来ました。考えてみると、私たちの周りはさまざまなイメージで満ちあふれています。多様化する価値観、目まぐるしく移り変わる流行。その中に秘められた可能性をひとつの形としてつくりあげること、それは現代に生きる私たちにとっても変わることのない大きな喜びです。
21世紀を目前にして、私たちは、大きな転換期ともいうべき時代を迎えています。
国際的な社会情勢の急激な変化、為替の変動、貿易摩擦、エネルギー問題など、経済環境は年々予測が難しくなっています。また、エレクトロニクス技術の発達によって、社会のすみずみまでマイクロコンピュータ制御がいきわたり、あらゆる分野で自動化が進んでいます。このような時代にあって、ものの価値、ものをつくることの意味も大きく変わろうとしています。
ものを「つくる」時代から、新たな価値を「創造」する時代へ。ものを「つくる」ためにテクノロジーの粋を集めてそそぎ込むだけでは、もはや十分とは言えません。激動する時代の流れを素早く読みとって、新たな価値、新たな意味を探し出し、誰よりも早く未知の果実をひとつの形に実らせる力が必要です。
CIM(Computer Integrated Manufacturing)も、これからの経営戦略の成否を決定する重要な鍵となります。最新のテクノロジーと情報化という概念の構築、そして人間の創造力を統合してはじめて、このマネージメントは実現します。
このように情報化の中で生まれる新たなる未知の果実。私達は常に進化し続ける状況に対応すべく、柔軟な発想と絶え間ない最先端技術とを追求し続けています。そしてそれゆえに、TAMの企業活動の基本は、それぞれの個人がいかに快適に、いかに創造的に仕事に向かうことができるか、ということなのです。この環境の中で育った種子は、新たな果実となり、既存の価値や関係を革新していくことでしょう。未来のプロダクツ、未来のマネジメントを思考するすべての人へ、私達は新たな価値創造を提言します。
TAM

戦略的マネジメントを目指して

私たちは考えます

新たな価値を見いだすこと、未知のイメージを具現化して形にし、消費者に提供すること。それは現代の企業に課せられた使命だと言えるでしょう。

時代の要請に応じて企業のかたちは進化します。生産・物流・研究開発・営業、それぞれの部門が個別に効率化、情報化を追求する「製品」至上主義の企業から、経営戦略をサポートする統合化された情報システムを持つ「情報」武装企業へ。体質の転換なくして生き残ることはできません。CIM(コンピュータによる統合生産)は、企業の未来の姿です。

私たちは提案します。

CIM実現の手段は千差万別。企業の数だけCIMがあるといっても過言ではありません。しかし、その求めるところは同じです。各部門が必要とする機能を独立して実現すること、部門間のスムーズな情報の流れを確立すること、またどの部門からでもCIM化できること。TAMは企業の実情にあったCIM化を推進するため、3つのメインテクノロジーと3つのインターフェーステクノロジーを提案します。

私たちは提供します

TAMS(Technology of Automated Manufacturing Solutions)はTAMの提案するCIMコンセプトです。設計にTAMS-Ⅰ、製造にTAMS-Ⅱ、計画にTAMS-Ⅲ。まず、導入できるところから始める。TAMは自立分散化した機能パッケージでCIM化を支援します。

価値の創造の基盤となる3つの条件を満たすシステムを提供します。

今、企業には価値を自ら創造することが求められています。新しい時代には、新しい製品、新しい価値を。企業の生みだす製品が消費を促し、消費はまた新たなニーズを生み、お互いに成長していきます。

しかし、創造を現実にし、ひとつの製品として具現化するとき、そこにはどうしても避けて通ることのできない問題があります。つまり、いかにして時間を最大限に活用するか、いかにして高いクォリティを保証するか、いかにしてコストを最低限に抑えるか、そしてこの3つを同時に満たすには、どうすべきかという問題です。

顧客は品質の良い製品を、それに見合った価格で、欲しい時に欲しいだけ手に入れたいと考えています。製品を提供する企業は、製品の品質をいっそう改善し、原価を抑え、ちょうど良いタイミングで顧客に提供できるよう効率的なスケジュールを組まなければならないと考えています。製品の需要と供給のタイミング、製品のクォリティ、製品の適切なコスト。顧客の要求と企業の求めるものはひとつの鏡の表と裏です。

私たちは、価値の創造の基盤となる3つの条件を満たすシステムを提供します。

Customer`s View



TAM`S Philosophy



Management`s View



エンジニアリングデータベースを核とした3つのテクノロジーと3つのインターフェーステクノロジーでCIMを実現します

現代の企業がこれからの時代を生き残るためには、効果的な生産計画の立案、品質の改善、生産コストの低減という3つの条件を満たさなくてはなりません。それを実現するのがTAMの提案するCIMシステムです。

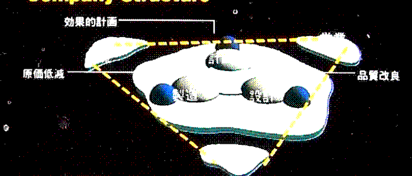
製造業を構成するのは、工場とそれを取り巻く物流、営業、研究開発の3部門。そして工場を構成するのは製造・計画・設計の3エリア。これまで、製造業では部門ごと、エリアごとにオートメーション化、コンピュータ化をはかってきました。しかし、これだけではもはや十分とは言えません。

これからの企業システム構築のキーは情報とそのインターフェースにあります。各部門、各エリアがそれぞれ独自の機能を果たしながら、他の部門、他のエリアにリアルタイムで情報を伝達する。このときはじめてシステムは全体として有機的に機能するといえるのです。

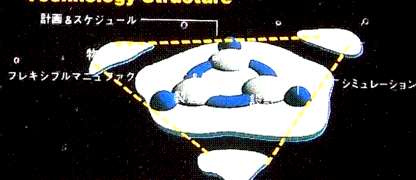
TAMはエンジニアリングデータベースを核とした3つのメインテクノロジーと3つのインターフェーステクノロジーでCIMを実現します。

3つのメインテクノロジーは、製造、計画、設計の3つのエリアを直接サポートし、他部門と工場を結びます。3つのインターフェーステクノロジーは、製造、計画、設計の3つのエリア相互にリアルタイムで情報を伝えます。

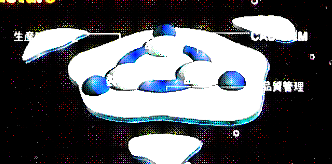
Company Structure



Plant Management Technology Structure



Plant Technology Structure



CIMへの具体的なアプローチは、企業の実情に応じどこからでも取り組めるものでなければならないとTAMは考えます

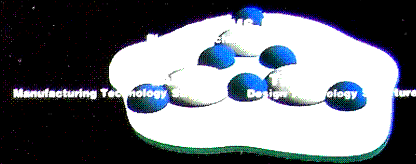
CIMの最終的な目標は投入資本に対する利益の増加です。まず、製品の多様化、納期の短縮といった顧客ニーズに対応することによって、売上増加、利益の増加をはかる必要があります。同時に、これに伴うコスト増を吸収するため、製造方法の合理化をいっそう押し進めなければなりません。また、在庫の削減、設備稼働率の向上によって投入資本そのものも改善しなければならないでしょう。

では、CIMの導入にはどこから手をつけたらよいのでしょうか? 企業の規模・業種・実績に応じてさまざまな答えがあるはずです。

TAMSは3つの製品パッケージで構成されています。直接工数の削減には、開発/設計をコンピュータで支援するTAMS-Ⅰ、品質向上、在庫の低減には品質管理と生産時点管理を中心とするTAMS-Ⅱ、市場ニーズに合ったタイミングでの工場スケジュール作成にはTAMS-Ⅲ。3つのシステムで統合環境を作ります。3つの製品は単独で導入することも、またTAMS-ⅠとTAMS-Ⅱだけを導入し、段階的にCIM化することも可能です。

TAMSなら、導入する企業の実情に応じたCIMシステムが構築できます。

TAMS Plant Technology Structure



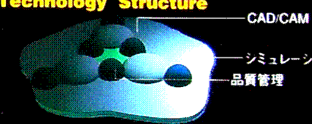
Technology of Automated Manufacturing Solutions TAMSはTAMの製品コンセプトです

TAMの提供する3つの製品は、設計・製造・計画の3分野についての独立した製品パッケージであると同時に、LANによって、これらをリアルタイムに統合したシステムとして統合制御できるようにデザインされています。

設計支援にTAMS-Ⅰ

コンピュータを使った設計システム(CAD/CAM)によって製品を設計し、NCデータを自動作成するだけでなく、設計モデルデータを品質管理(CAQ)、製品シュミレーション(CAE)システムと連動させ、3つのテクノロジーを統合して設計エリアのCIM化を支援します。

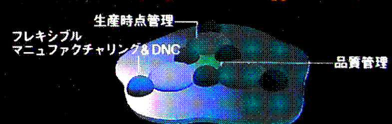
Design Technology Structure



製造支援にTAMS-Ⅱ

工場で発生するあらゆる生産情報をリアルタイムで管理し(POP)、製造過程(FM & DNC)に製造支援データを送り、生産を直接制御します。POPは品質管理(CAQ)システムと、またCAQはFM & DNCと直接連動して高い品質を保証し、3つのテクノロジーを統合して製造エリアのCIM化を支援します。

Manufacturing Technology Structure



計画支援にTAMS-Ⅲ

工場のスケジュール管理システム(PP & DS)から、生産時点管理(POP)システム、設計(CAD/CAM)システムに指示を与え、3つのテクノロジーを統合して計画エリアのCIM化を支援します。

Planning Technology Structure



会社概要

創業	昭和49年2月
設立	昭和58年7月
資本金	5,000万円
役員	代表取締役社長 市原静司
事業内容	CAD/CAM、CIMのシステムインテグレーション、及びコンサルテーション
年商	27億円(平成2年3月実績)
社員数	90名

尚、詳しい資料をご請求の方は、下記までご連絡ください。

タム株式会社
〒160東京都新宿区西新宿4丁目15番7号
後楽園新宿ビル7階
TEL 03(299)8811 総務部 吉村、高島

MIFES VZ-Editor 標準キー設定、機能番号比較一覧表

機能番号（VZ-Editorではコマンドコード）をマクロ中に記述することで各機能を実現することができます。（MIFES Ver.5.0、VZ Ver.1.5xに対応）

■ファイル関係の操作

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
テキストをセーブしてクローズ	[f・1],[E]	84	—	—
テキストのセーブ	[f・1],[S]	85	[ESC],[S]	#75
テキストをクローズ	[f・1],[Q]	86	[ESC],[C]	#73
テキストを表示右マージン通りセーブ	[f・1],[M]	144	—	—
テキストの編集のやり直し	[f・1],[O]	87	[ESC],[U]	&07
テキストの２重化	[f・1],[D]	88	[ESC],[D]	&06
テキストのパス名変更	[f・1],[R]	89	—	—
ファイル名の変更	—	—	[ESC],[P]	#82
全ファイルをセーブしてエディタの終了	[f・1],[A]	90	—	—
全ファイルをクローズ	—	—	[ESC],[X]	&09
エディタの終了	—	—	[ESC],[Q]	#77
全文書を破棄してエディタの終了	[f・1],[0]	91	[GRPH]+[X]	&27
カレントテキストにファイルの挿入	[f・1],[I]	92	[ESC],[A]	#76
新規ファイルのオープン	[f・2],[N]	60	[ESC],[N]	#72
ファイルのリードオンリーオープン	—	—	[ESC],[R]	#71
ファイルのオープン	—	—	[ESC],[O]	#70
クローズ、オープン	—	—	[ESC],[L]	#74
プロファイル情報のセーブ	—	—	[ESC],[W]	#83

■カーソルの移動

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
カーソルを１行上に移動	[↑]	5	[↑]	#05
カーソルを１行下に移動する	[↓]	24	[↓]	#06
カーソルを１文字左に移動	[←]	19	[←]	#07
カーソルを１文字右に移動	[→]	4	[→]	#08
スクリーンを２行ロールダウン	[SHIFT]+[↑]	145	—	#33
スクリーンを２行ロールアップ	[SHIFT]+[↓]	146	—	#34
カーソルを行の左端に移動する	[SHIFT]+[←]	54	[CTRL]+[←]	#11
カーソルを行の右端に移動する	[SHIFT]+[→]	55	[CTRL]+[→]	#12
スクリーンのロールアップ	[ROLL UP]	3	[SHIFT]+[↓]	#36
スクリーンのロールダウン	[ROLL DOWN]	18	[SHIFT]+[↑]	#35
スクリーンのスムースアップ	—	—	[ROLL DOWN]	#37
スクリーンのスムースダウン	—	—	[ROLL UP]	#38
カーソルを１語前方に移動	[CTRL]+[F]	6	[SHIFT]+[←]	#09
カーソルを１語後方に移動	[CTRL]+[A]	1	[SHIFT]+[→]	#10
スクリーンを１行分ロールアップ	[CTRL]+[Z]	26	—	—
スクリーンを１行分ロールダウン	[CTRL]+[W]	23	—	—
カーソルを行の左端（または右端）に移動	—	2	—	—
カーソルを次のタブストップ位置へ移動	—	17	—	—
カーソルをスクリーンの上端へ移動	—	—	[CTRL]+[↑]	#39
カーソルをスクリーンの下端へ移動	—	—	[CTRL]+[↓]	#40
カーソルをスクリーンの左端へ移動	—	—	[CTRL]+[Q],[[]	&12
カーソルをスクリーンの右端へ移動	—	—	[CTRL]+[Q],[]]	&13
カーソルを指定行番号へのジャンプ	[f・3],[G]	76	[SHIFT]+[f・2],[J]	#45
カーソルをテキスト先頭へジャンプ	[f・3],[T]	77	[CTRL]+[ROLL DOWN]	#41
カーソルをテキスト最後へジャンプ	[f・3],[B]	78	[CTRL]+[ROLL UP]	#42
タグジャンプ	[f・10]	41	[SHIFT]+[f・10]	&18
対応する括弧の検索	—	67	[SHIFT]+[HOME CLR]	#58
カーソルを論理的な行末まで移動	—	68	—	—
現在のカーソル位置を登録	—	116	[CTRL]+[Q],[M]	#44
カーソルをシステムマークへジャンプ	[CTRL]+[Q],[J]	114	[CTRL]+[Q],[P]	#43
カーソルをマークnへジャンプ	[f・3],[n]	—	[CTRL]+[Q],[n]	&17
行マークの登録	[CTRL]+[J],[n]	51	[CTRL]+[K],[n]	&16
タブ挿入またはタブ文字での置換	—	115	—	—
ページングモード	—	—	[SHIFT]+[INS]	#30

■文字（列）の挿入／削除

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
改行文字の挿入または行の分割	[RETURN]	13	[RETURN]	#04
タブ文字の挿入	[TAB]	9	[TAB]	#21
カーソル位置の１文字を削除	[DEL]	7	[DEL]	#15
カーソル直前の１文字を削除	[BS]	8	[BS]	#14
１行削除→削除文字列バッファへ移動	[CTRL]+[Y]	25	—	—

"[A],[B]"は[A],[B]を順次押す,"[A]+[B]"は[A]を押しながら[B]を押す,[n]は任意の数字キーを押すことを表します.

■文字（列）の挿入／削除

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
カーソル位置から行末まで削除	[CTRL]+[K]	11	[CTRL]+[DEL]	#19
行頭からカーソル直前までを削除	[CTRL]+[U]	21	[CTRL]+[BS]	#18
カーソル位置の１語削除	[CTRL]+[T]	20	—	—
カーソル位置から語の先頭まで削除	—	—	[SHIFT]+[BS]	#16
カーソル位置から語の終りまで削除	—	—	[SHIFT]+[DEL]	#17
直前の削除のアンドゥ	[CTRL]+[Q],[U]	111	[CTRL]+[U]	#20
ペーストのアンドゥ	[CTRL]+[Q],[U]	111	[CTRL]+[K],[U]	&20
カーソルのある行を２重化	[SHIFT]+[f・10]	49	—	—
カーソル行の下に１行挿入	[SHIFT]+[RETURN]	53	—	—
カーソル行の上に１行挿入	[CTRL]+[N]	14	—	—
削除文字列バッファの内容をペースト	[CTRL]+[L]	12	—	—
削除文字スタックから１文字ペースト	[CTRL]+[P]	16	—	—
任意の文字（00H〜FFH）の入力	[CTRL]+[V]	22	[CTRL]+[P]	#22
疑似改行文字の入力	[CTRL]+[O]	74	—	—
１行削除→行カットバッファへ移動	—	75	—	—
行削除スタックから１行ペースト	[SHIFT]+[INS]	108	—	—
文字列の複写	—	—	[SHIFT]+[f・5]	#24
ファイル名の複写	—	—	[CTRL]+[Q],[N]	#25
カーソル行の複写	—	—	[CTRL]+[K],[D]	&19
カーソル位置へ挿入	—	—	[ESC],[I]	&08

■ブロックの移動／複写／削除

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
行単位でのテキスト選択操作を開始	[f・6]	37	—	—
文字列単位のテキスト選択操作を開始	[SHIFT]+[f・6]	47	—	—
選択したテキストをカットバッファへの移動	[f・7]	38	—	—
行・ブロック削除	—	—	[f・8]	#47
選択したテキストをカットバッファへの複写	[f・8]	39	—	—
行・ブロック複写	—	—	[SHIFT]+[f・9]	#50
行カットバッファをカーソル行の上に挿入	[f・9]	40	—	—
行・ブロック挿入	—	—	[f・9]	#48
文字列カットバッファの内容をペースト	[SHIFT]+[f・9]	48	—	—
ブロックモード	—	—	[f・10]	#46
矩形ブロックモード	—	—	[CTRL]+[K],[B]	&60
行・ブロック記憶	—	—	[SHIFT]+[f・8]	#49
スタックの消去	—	—	[SHIFT]+[f・2],[C]	#51
ブロックの先頭／最後	—	—	[SHIFT]+[f・2],[B]	#52
１行挿入	—	—	[SHIFT]+[RETURN]	#53
行の回復	—	—	[SHIFT]+[f・2],[U]	#54

■文字（列）の検索／置換

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
文字列の前方検索	[SHIFT]+[f・3],[F]	30	—	—
文字列の後方検索	[SHIFT]+[f・3],[B]	28	—	—
カレント検索文字列の前方検索	[f・5]	36	—	—
カレント検索文字列の後方検索	[SHIFT]+[f・5]	46	—	—
検索文字列の設定	—	—	[f・6]	#55
検索文字列の取得	—	—	[f・5]	#59
文字列の連続置換（確認あり）	[SHIFT]+[f・3],[Q]	79	—	—
文字列の連続置換（確認なし）	[SHIFT]+[f・3],[A]	80	—	—
文字列の連続置換（範囲指定）	[SHIFT]+[f・3],[R]	81	—	—
文字列の置換	—	—	[f・7]	#56
複数のテキストでの検索／置換	—	—	[SHIFT]+[f・6]	&21
最後に実行した文字列の連続置換を継続実行	[SHIFT]+[f・3],[C]	82	[CTRL]+[Q],[O]	#57
検索不成功時のマクロの中止／継続の切り替え	—	58	—	—
タグ情報の取得	—	—	[CTRL]+[K],[G]	#60
文字列の選択複写	—	—	[SHIFT]+[f・7]	#61
検索文字列の連続取得	—	—	[f・5]	&24
半角英文字を大／小文字変換	[CTRL]+[]]	29	[CTRL]+[K],[Z]	&69
１語大／小文字変換	—	—	[CTRL]+[Q],[U]	#23

■画面の操作

	MIFES 標準キー設定	MIFES 機能番号	VZ-Editor 標準キー設定	VZ-Editor 機能番号
スプリットの切り替え	[f・4]	35	—	—
ウィンドウの切り替え	—	—	[f・2]	#62
スプリット境界線の位置変更	[SHIFT]+[f・4]	45	—	—

機能番号（VZ-Editorではコマンドコード）をマクロ中に記述することで各機能を実現することができます．（MIFES Ver.5.0，VZ Ver.1.5xに対応）

	MIFES		VZ-Editor	
	標準キー設定	機能番号	標準キー設定	機能番号

■画面の操作（おもてより続き）

分割位置の変更	—	—	[SHIFT]+[f・4]	#65
カレントテキストと参照テキストの切り替え	[HOME CLR]	50	—	—
編集テキストの切り替え	—	—	[f・3]	#63
画面を非スプリット状態にする	[SHIFT]+[HOME CLR]	52	—	—
参照ウィンドウの１行ロールアップ	[SHIFT]+[ROLL UP]	129	—	—
参照ウィンドウの１行ロールダウン	[SHIFT]+[ROLL DOWN]	130	—	—
現テキストを２分割表示する	—	107	—	—
分割モードの切り替え	—	—	[f・4]	#64
テキストの比較	—	—	[SHIFT]+[f・3]	#69
ビューモード	—	—	[CTRL]+[Q],[V]	&26

■その他の編集操作

文字列の補完	—	—	[CTRL]+TAB]	&22
文字コード表示	—	—	[CTRL]+[K],[J]	&25
半角罫線モード	—	124	—	—
全角罫線モード	—	125	—	—
カーソル行のリフォーム	[CTRL]+[Q],[Q]	126	—	—
一時的右マージンの変更	—	123	—	—
ESCプリフィックス	[ESC]	27	[ESC]	#03
メモマークの挿入	[CTRL]+[Q],[M]	131	—	—
ヘルプの表示	[HELP]	10	[HELP]	&01

■マクロ操作

キーボードマクロの定義開始及び終了の指定	[CTRL]+[@]	0	—	—
キーボードマクロの実行	[CTRL]+[_]	31	[CTRL]+[_]	#02
キーボードマクロ文挿入	—	—	[CTRL]+[K],[_]	#81
外部マクロコマンドの実行	—	99	[ESC],[_]	&10
マクロ文の実行	—	—	[ESC],[@]	&11
現在のキーボードマクロをテキスト中心にペースト	—	104	—	—
カーソル行の内容をキーボードマクロに登録する	—	141	—	—
直前に実行した２ストローク操作の再実行	[CTRL]+[B]	137	—	—
直前のコマンドの実行	—	—	[CTRL]+[[]	#01
指定した英字バッファをキーボードマクロに複写	—	135	—	—
最後に実行した外部マクロコマンドの再実行	—	136	—	—
行カットバッファの内容をプリンタに出力	[CTRL]+[Q],[P]	139	—	—
行カットバッファ表示右マージン通りにプリンタに出力	[CTRL]+[Q],[O]	143	—	—
再カスタマイズ	—	—	[CTRL]+[K],[_]	#80

■状態の表示変更

無操作（何もしない）	—	56	—	—
挿入／上書モードの切り替え	[INS]	15	[INS]	#13
カーソル位置の文字コードの表示	—	69	—	—
スプリット形状の上下／左右の切り替え	[SHIFT]+[f・1],[S]	62	—	—
行番号の表示／非表示の切り替え	[SHIFT]+[f・1],[N]	63	—	—
桁位置ゲージの表示／非表示の切り替え	[SHIFT]+[f・1],[C]	64	—	—
オートインデントのON/OFFの切り替え	[SHIFT]+[f・1],[I]	66	—	—
両画面スクロールのON/OFFの切り替え	[SHIFT]+[f・1],[D]	70	—	—
ファイル比較モードのON/OFFの切り替え	[SHIFT]+[f・1],[E],[C]	117	—	—
アンダーラインカーソルのON/OFFの切り替え	[SHIFT]+[f・1],[U]	73	—	—
改行文字の表示／非表示の切り替え	[SHIFT]+[f・1],[R]	95	—	—
タブ文字の表示／非表示の切り替え	[SHIFT]+[f・1],[T]	96	—	—
検索時の英大小文字の区別／同一視の切り替え	[SHIFT]+[f・1],[F]	98	—	—
ディフォルトハードタブの変更	[SHIFT]+[f・1],[E],[H]	121	—	—
ディフォルト右マージンの変更	[SHIFT]+[f・1],[G]	119	—	—
キャラクターコード表の表示と入力	[SHIFT]+[f・2],[T]	72	—	—
行カットバッファの内容の確認	[SHIFT]+[f・1],[1]	100	—	—
文字列カットバッファの内容の確認	[SHIFT]+[f・1],[2]	101	—	—
削除文字列バッファの内容の確認	[SHIFT]+[f・1],[3]	102	—	—
検索文字列の履歴管理を初期化	—	103	—	—
ページスクロール動作のジャンプ／スムースの切り替え	[SHIFT]+[f・1],[E],[R]	65	—	—
[EOF]記号の表示／非表示の切り替え	[SHIFT]+[f・1],[E],[E]	118	—	—
クリック音のON/OFFの切り替え	[SHIFT]+[f・1],[E],[K]	109	—	—

"[A],[B]"は[A],[B]を順次押す,"[A]+[B]"は[A]を押しながら[B]を押す,[n]は任意の数字キーを押すことを表します．

	MIFES		VZ-Editor	
	標準キー設定	機能番号	標準キー設定	機能番号

■状態の表示変更

マクロのシングルステップ実行のON/OFFの切り替え	—	71	—	—
バックアップファイルの作成／未作成の切り替え	[f・1],[B]	120	—	—
画面モードの切り替え	[SHIFT]+[f・1],[E],[2]	94	[SHIFT]+[f・2],[L]	#66
大文字変換モード／小文字変換モードの切り替え	[SHIFT]+[f・1],[E],[A]	97	—	—
ファイル名表示のフルパス／ファイル名のみの切り替え	—	110	—	—
高速スクロール動作の切り替え	[CTRL]+[Q],[W]	59	—	—
英字バッファの一覧を表示	—	61	—	—
カレントドライブ及びカレントディレクトリの変更	—	112	—	—
ディスク／メモリ情報の表示	—	113	—	—
上書モードの動作（桁対応／文字コード）切り替え	[SHIFT]+[f・1],[E],[O]	127	—	—
ガイドラインの表示（行桁表示／文字対応）切り替え	[SHIFT]+[f・1],[E],[G]	128	—	—
行番号表示の論理行番号／表示行番号の切り替え	[SHIFT]+[f・1],[E],[N]	133	—	—
ページマークの表示行数を変更	—	134	—	—
マクロ用ユーザ変数の表示と変更	—	57	—	—
正規表現機能の有効／無効の切り替え	[SHIFT]+[f・3],[X]	106	—	—
画面のガイド表示のON/OFFの切り替え	[CTRL]+[Q]	138	—	—
バーチカルラインカーソルのON/OFFの切り替え	[CTRL]+[V]	140	—	—
行端カーソル制御の止まる／止まらない切り替え	[CTRL]+[A]	142	—	—
スクロールマージンの変更	—	148	—	—
フリーカーソルモードのON/OFFの切り替え	[SHIFT]+[f・1],[E],[F]	149	—	—
リードオンリーモード	—	—	[SHIFT]+[f・2],[R]	#67
インデントの変更	—	—	[SHIFT]+[f・2],[I]	#68
タブの切り替え	—	—	[CTRL]+[K],[I]	&15
画面のカーソル位置変更	—	—	[HOME CLR]	&14

■子プロセスの実行

DOSコマンドの実行	[SHIFT]+[f・2],[S]	83	[ESC],[E]	#78
標準エラー出力を指定してDOSコマンドの実行	[SHIFT]+[f・2],[E]	105	—	—
一覧表示機能付きDOSコマンドの実行	[SHIFT]+[f・2],[O]	93	—	—
ヒストリー機能付き子プロセス	[CTRL]+[Q],[H]	132	—	—
DOSコマンドラインでのファイラー	—	—	[SHIFT]+[RETURN]	#98
DOSコマンドラインでのエイリアス一覧	—	—	[CTRL]+[W]	#96
DOSコマンドラインでのコマンドの展開再表示	—	—	[CTRL]+[RETURN]	#97
DOSコマンドラインでのDOS画面出力の参照	—	—	[SHIFT]+[ESC]	#95
DOS画面出力の参照	—	—	[SHIFT]+[ESC]	#79

■ファイラー

ファイラーでの選択←	—	—	[SHIFT]+[←]	#88
ファイラーでの選択→	—	—	[SHIFT]+[→]	#89
ファイラーでの選択↑	—	—	[SHIFT]+[↑]	#90
ファイラーでの選択↓	—	—	[SHIFT]+[↓]	#91
ファイラーでの先頭／最後	—	—	[BS]	#92
ファイラーでのウィンドウの切り替え	—	—	[DEL]	#93

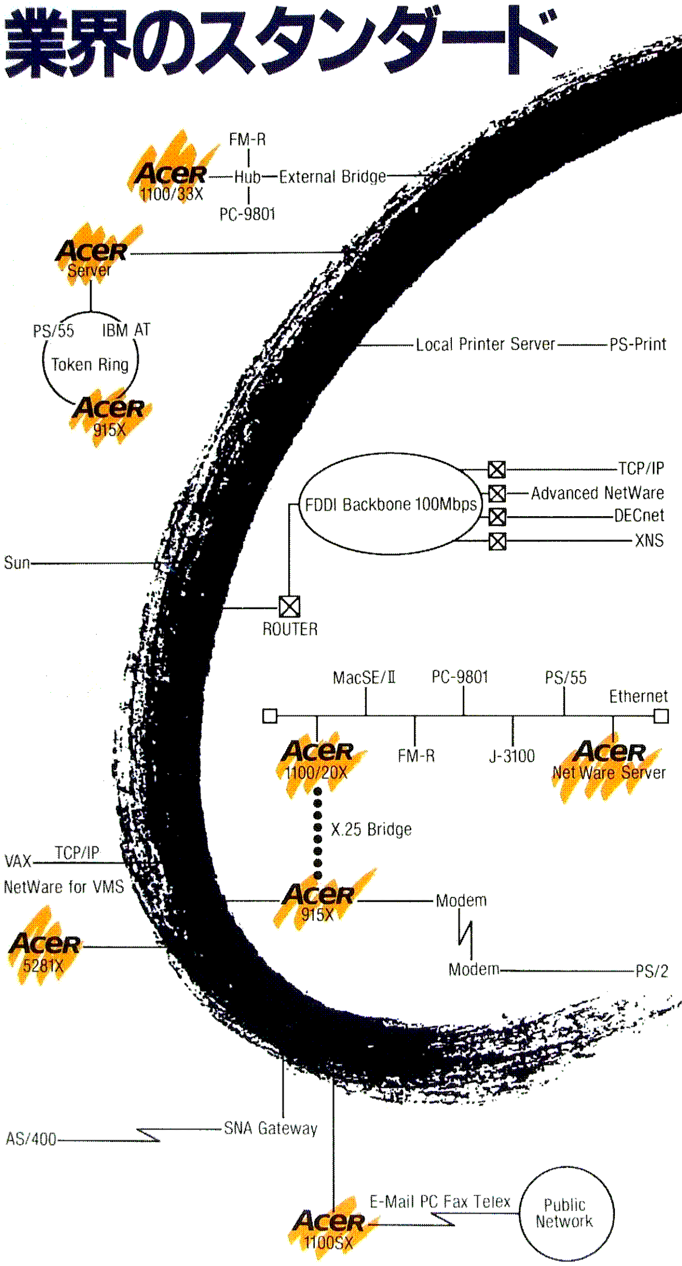
■メニューのオープン

ファイルメニュー	[f・1]	32	[f・1]	&02
テキスト切り替えメニュー	[f・2]	33	—	—
ジャンプメニュー	[f・3]	34	—	—
ステータスメニュー	[SHIFT]+[f・1]	42	—	—
拡張ステータスメニュー	[SHIFT]+[f・1],[E]	122	—	—
モードメニュー	—	—	[SHIFT]+[f・1]	&03
子プロセスメニュー	[SHIFT]+[f・2]	43	—	—
置換メニュー	[SHIFT]+[f・3]	44	—	—
サブメニュー	—	—	[SHIFT]+[f・2]	&04
マクロメニュー	—	—	[ESC],[M]	&05
作表／ツールメニュー	—	—	[ESC],[H]	&70

NetWare

Acer
The Word for Value

パソコンLANにおける業界のスタンダード



THE ARCHITECTURE EXTENDED-AX

世界のパソコンLAN市場をリード

米国Novell社の"NetWare"は、世界のパソコンLAN市場において60%以上の圧倒的なシェアをもつLAN OSです。

このNetWareに日本語が付加されたことによってACERなどのIBM-PC(AT互換機)はもちろんのことAXマシン、PC-9801、FM-R、PS-55、J-3100などの国産PCへの対応も広がっており、日本語・英語両環境でのネットワーク構築を可能といたしました。

SFT NetWareの特徴

SFT NetWareは、ミニコンやメインフレームでしか提供できなかったさまざまな機能をパソコンベースで実現することができます。他のLANシステムと比較した場合の大きな特徴の一つにシステム・フォルト・トレラント（SFT）機能があげられます。例えばファイルサーバーに2台のハードディスクを装着させることによって"ディスクの二重化"が可能であり、常に同じ情報が2台のハードディスクに格納されるため、万一メインのハードディスクに故障が生じても、サブ側のハードディスクが動作し続け通常業務に影響を与えることはありません。(ディスク・デュープレックス機能)

また、書き込みのたびに、そのセクターを読み出し、セクターに不良があるとわかったら、以後そこを使わないようにする不良セクター対策（Hot Fix機能）やOSにとって重要なディレクトリー・テーブルとファイル・アロケーション・テーブルについては、それぞれ別の場所に書き込んでおき、読み出しの際には、コピーと比較してエラー検査をします。(ディレクトリー構造の二重化、ファイル・アロケーション・テーブルの二重化など…)

以上のようにセキュリティと障害対策が他社製品に比べて非常に強力であることがSFT NetWareの特徴といえます。

●NetWareとTCP/IPの共存が可能

TCP/IPプロトコル、PC/NFS機能によりUNIXマシンおよびSUNなどのワークステーションをサーバーにするシステムをサポート。これによってNetWareとTCP/IPソフトが同時に共存可能になりました。

●D-Link　ポケット・イーサネットアダプター　新発売

J-3100の様なラップトップAT互換機のパラレルポートに接続できるポケット型イーサネットアダプターです。(NetWare対応)

世界の舞台で磨かれた

Acerファイル・サーバー Acer1100/33 (PC/AT互換機)をベース



- i80386(33MHz)CPU,RAM 4MB
- 5.25インチ(1.2MB)FDD×1基、
- 100MB～680MB HDD最大2基 実装可能
- 14インチモノクロモニター、モニターI/Fカード
- 16ビットバス用 Ethrnetカード1枚 標準装備
- フロアスタンド他

※Net WareはNOVELL Inc.の登録商標です。

丸文株式会社

東京都中央区日本橋大伝馬町8-1 丸文ダイヤビル 〒103 第1事業本部 海外営業部C&Cシステムグループ TEL (03)639-9897(ダイヤルイン)

〈資料請求番号P03〉

雑誌コード 17777-01 T4911777701983 〈資料請求番号P04〉